



UNIVERSIDAD POLITÉCNICA DE MADRID

FACULTAD DE INFORMÁTICA

# NUEVOS MÉTODOS PARA ANÁLISIS VISUAL DE MAPAS AUTO-ORGANIZATIVOS

TESIS DOCTORAL

AUTOR: MANUEL RUBIO SÁNCHEZ  
LICENCIADO EN INFORMÁTICA

2004





*datsi*



DEPARTAMENTO DE ARQUITECTURA Y TECNOLOGÍA  
DE SISTEMAS INFORMÁTICOS

LABORATORIO DE COMUNICACIÓN ORAL  
*“Robert Wayne Newcomb”*

FACULTAD DE INFORMÁTICA

# NUEVOS MÉTODOS PARA ANÁLISIS VISUAL DE MAPAS AUTO-ORGANIZATIVOS

AUTOR: MANUEL RUBIO SÁNCHEZ  
LICENCIADO EN INFORMÁTICA

DIRECTORES:  
VÍCTOR GIMÉNEZ MARTÍNEZ, DOCTOR EN INFORMÁTICA  
KARIM DROUCHE, DOCTOR EN MATEMÁTICAS

2004





Tribunal nombrado por el Magfco. y Excmo. Sr. Rector de la Universidad Politécnica de Madrid, el día *29 de marzo de 2004*.

Presidente     *D. Pedro Gómez Vilda*

Vocal             *D. Elmar Wolfgang Lang*

Vocal             *D. José Manuel Ferrández Vicente*

Vocal             *D. Carlos García Puntonet*

Secretario     *D<sup>a</sup>. Consuelo Gonzalo Martín*

Suplente         *D. Juan Bautista Castellanos Peñuela*

Suplente         *D. Carlos Antonio Hernández Espinosa*

Realizado el acto de defensa y lectura de la Tesis el día *28 de abril de 2004* en la Facultad de Informática.

Calificación:    *Sobresaliente Cum Laude por unanimidad*

EL PRESIDENTE

LOS VOCALES

EL SECRETARIO



# Agradecimientos

Quisiera agradecer a todos los que han hecho posible la realización de la presente tesis doctoral su apoyo y colaboración durante los años que he utilizado para su confección. Es difícil mencionar en pocas líneas a todos los que me han ayudado a la realización de este trabajo, de forma que pido disculpas a los que no son mencionados explícitamente.

En concreto, quisiera comenzar por Víctor Giménez y Karim Drouiche, mis directores de tesis, quienes han alentado mi interés por la investigación y han fomentado el desarrollo de la tesis, aconsejándome en todas mis dudas. Estoy muy agradecido por la oportunidad que me ha brindado Víctor Giménez de participar en sus clases y sesiones en congresos donde he podido divulgar mi trabajo, obteniendo un reconocimiento que ha sido una fuente de motivación fundamental para el desarrollo de esta tesis. Me gustaría agradecer asimismo el apoyo y trato de Karim Drouiche, con quien espero elaborar alguna teoría interesante próximamente, por darme la oportunidad de disfrutar de una valiosa experiencia tanto a nivel profesional como personal.

Quisiera dar las gracias al personal del Departamento de Arquitectura y Tecnología de Sistemas Informáticos, en especial a Pedro Gómez y a Víctor Nieto por darme la oportunidad de trabajar con ellos durante tantos años, por su apoyo y valiosos consejos, y sobre todo por su flexibilidad para permitirme explorar varios campos dentro de mi labor como investigador, algunos no tratados en el pasado por el departamento, tomando al pie de la letra la siguiente frase de Pedro: “la tecnología es muy amplia”. He aprendido mucho colaborando con vosotros y espero seguir haciéndolo en el futuro.

No debo olvidarme de Jose María García, Francisco J. Rodríguez, Francisco Díaz y otros compañeros del Laboratorio de Comunicación Oral *Robert Wayne Newcomb*, que me han incentivado, aconsejado y animado en mi tarea doctoral. Quisiera dar las gracias también a Marco Mancini y al personal del laboratorio

LPTM de la universidad de Cergy-Pontoise por su calurosa acogida durante mis dos estancias en Francia y con con quienes aprendí tanto. También debo dar las gracias a Bożena Pająk, no sólo por haberme ayudado a corregir el escrito, sino por haberme dado una auténtica lección de castellano.

Y finalmente a mis padres y hermano, que en todo momento supieron comprender mis inquietudes personales y mis necesidades de tiempo imprescindibles para la realización de un trabajo de estas características.

A todos, muchas gracias.

*Manuel Rubio Sánchez*

# Prefacio

Los métodos presentados en esta tesis han sido diseñados para facilitar el análisis exploratorio de datos con mapas auto-organizativos de Kohonen. Los trabajos están orientados principalmente hacia una interpretación mejorada de éstos, y por consiguiente de la distribución de datos con la cual han sido entrenados. En este sentido, cabe destacar que este trabajo se centra, ante todo, en *MAPAS AUTO-ORGANIZATIVOS*. La mayoría de los estudios de investigación llevados a cabo previos a la documentación final de esta tesis están relacionados con este tipo de red neuronal artificial, abarcando campos como la inicialización, clasificación automática, medidas de bondad, variantes dinámicas y *VISUALIZACIÓN*. Dados los resultados y la originalidad de los algoritmos propuestos relacionados con la visualización, decidí finalmente orientar la tesis en esa dirección.

Esta tesis no debe ser considerada como libro de texto de análisis exploratorio de datos ni de redes neuronales artificiales. Tampoco es una recopilación de técnicas, más o menos adaptadas, aplicadas a un problema real. El estudio realizado es de carácter teórico, donde varias ideas originales y novedosas son propuestas, dentro de un marco del estado del arte. Aunque varios experimentos utilizan datos reales de habla, los algoritmos presentados han sido probados principalmente con distribuciones de datos de simulación, algunas especialmente difíciles de tratar por métodos clásicos de análisis de datos, clasificación o reconocimiento de formas. De esta manera, se pretende hacer hincapié en aspectos no tratados por los métodos clásicos de visualización de mapas auto-organizativos, con el objetivo de poder detectar más detalles de los datos y forzarnos a ver lo que nunca esperábamos, dentro de nuestro papel de detectives en el campo del análisis exploratorio de datos.

Los algoritmos (implementados con la herramienta de programación Borland C++ Builder 3.0. y Matlab) son capaces de generar por sí mismos imágenes con

las cuales es posible analizar características importantes de distribuciones de datos. Sin embargo, sería un error utilizarlas exclusivamente como técnicas óptimas de análisis. Los nuevos algoritmos deben ser combinados con otras técnicas de visualización de mapas auto-organizativos y por supuesto con otras técnicas de análisis exploratorio de datos y visualización estadística.

Por otro lado, es necesario tener en cuenta que los análisis llevados a cabo están basados en imágenes mostradas en la pantalla de un monitor determinado. Aunque los colores en dicho monitor no coinciden exactamente con los impresos en este trabajo, esto no debe ser crítico para entender la filosofía y los resultados de los métodos descritos. En cualquier caso, el lector interesado puede solicitarme las imágenes originales a través de email<sup>1</sup>.

Dado que prácticamente la totalidad de la literatura sobre los temas tratados está escrita en lengua inglesa, he traducido varios términos siguiendo un criterio personal, indicando la correspondencia entre las definiciones españolas e inglesas en el Apéndice de la tesis. Por último, las referencias a páginas de Internet en la bibliografía están actualizadas a fecha de 9 de diciembre de 2003.

Como todo autor, he intentado escribir esta tesis de manera que le resulte amena al lector, lo incite a continuar las investigaciones y a aplicar los métodos propuestos con datos reales. Por último, animo a todo aquel interesado a ponerse en contacto conmigo para comentar cualquier cuestión relacionada con este trabajo o con futuros trabajos relacionados con el mismo.

*Manuel Rubio Sánchez*

---

<sup>1</sup>manolo@junipera.datsi.fi.upm.es, manuelrubio2000@telefonica.net

# Resumen

El mapa auto-organizativo (MAO) es un tipo de red neuronal artificial competitiva y no-supervisada. Ha sido utilizado tradicionalmente en tareas de ingeniería como herramienta de clasificación automática (*clustering*) y especialmente en tareas relacionadas con el análisis exploratorio de datos y la minería de datos, ya que su propósito principal es la visualización de relaciones no-lineales de datos multidimensionales. Sin embargo, a pesar de la importancia de la tarea de visualización, las técnicas gráficas para analizar MAO no son abundantes en la literatura.

Esta tesis presenta varias técnicas nuevas que complementan, mejoran y facilitan el análisis visual de MAO de Kohonen, tanto desde el punto de vista del análisis exploratorio de datos, como desde el punto de vista de comprender el proceso de adaptación del MAO a una distribución de datos.

La motivación para desarrollar técnicas de visualización nuevas surge por los siguientes motivos: la relativa carencia de métodos destinados a la importante tarea de visualización, la necesidad de analizar MAO con diferentes métodos, la necesidad de mejorar varios métodos descritos en la literatura y la posibilidad de innovar desarrollando nuevas estrategias de visualización. De esta manera, se ha hecho hincapié en desarrollar técnicas generalmente no utilizadas con anterioridad en un intento por superar limitaciones de varios métodos descritos en la literatura.

El primer nuevo método denominado “método de semejanza de triángulos” consiste en una estrategia de interpolación geométrica donde los patrones de una distribución de entrada son proyectados a un espacio de observación continuo. Está basado en la preservación de la semejanza geométrica entre varios triángulos formados por un patrón y dos vectores de referencia del MAO en el espacio de los datos, y por un punto candidato y las dos correspondientes neuronas en el espacio de observación. El método encuentra la proyección minimizando una función de coste que mide distancias o errores entre varios triángulos. El método supera

notablemente a otras estrategias de interpolación descritas en la literatura. Puede proyectar todos los datos de manera no-lineal, resulta adecuado cuando el tamaño del MAO es pequeño, es robusto y puede describir adecuadamente ciertos tipos de distribuciones difíciles de visualizar con la mayoría de métodos de visualización.

Varios métodos de visualización de MAO generan imágenes monocromáticas las cuales son analizadas individualmente y aportan información específica sobre los datos. Se propone una estrategia para facilitar la labor del analista a la hora de combinar la información de varios métodos mediante una simple superposición de imágenes basada en un modelo aditivo de colores. Las imágenes son definidas con colores diferentes y combinadas mediante una simple suma de sus componentes de color. Las imágenes resultantes son más completas y robustas, especialmente cuando las imágenes a combinar aportan el mismo tipo de información. El estudio llevado a cabo se centra principalmente en la combinación de matrices de distancias con histogramas de datos.

Una alternativa a las matrices de distancias, que generan imágenes monocromáticas y son los métodos más populares para visualizar la estructura de *clusters* de los datos, consiste en emplear estrategias que ilustren los diferentes *clusters* mediante colores diferentes. Una de estas estrategias consiste en utilizar modelos de contracción de neuronas. Se presenta un eficiente método de contracción, el “algoritmo de agrupación de neuronas”, cuya estructura y filosofía es similar a la del algoritmo de entrenamiento de los MAO, donde los conceptos han sido invertidos para actualizar las posiciones de las neuronas en un mapa continuo en vez de los propios vectores de referencia del MAO. De esta manera, las neuronas son atraídas en el mapa en función de la distancia entre sus vectores de referencia en el espacio de los datos. Su principal ventaja es su bajo coste computacional que lo habilita para analizar MAO de tamaño elevado.

Finalmente, el trabajo propone una técnica alternativa basada en la visualización explícita en el mapa o espacio de observación de grafos que unen neuronas cuyos vectores de referencia se hallan próximos en el espacio de los datos, como son el árbol generador mínimo o el “grafo Hebbiano” creado con el principio de aprendizaje Hebbiano competitivo. Las imágenes resultantes ayudan a analizar la dimensión intrínseca de los datos en cada zona del mapa y aportan una medida visual e intuitiva de la preservación de la topología del MAO.



# Abstract

The self-organizing map (SOM) is a competitive, non-supervised artificial neural network. It is used traditionally in engineering tasks as a clustering tool and especially in exploratory data analysis and data mining related tasks, since its main purpose is the visualization of nonlinear relations of multidimensional data. However, despite the importance of the visualization task, graphical techniques for SOM analysis are not abundant in the literature.

This dissertation presents several new techniques that complement, enhance and facilitate SOM visual analysis, from an exploratory data analysis point of view and in an attempt to provide insight into the SOM's adaptation process to a data distribution.

The motivation to develop new techniques rises from: the relative lack of visualization methods, the need to analyze SOM with different methods, the need to enhance several methods described in the literature, and the possibility to innovate developing new visualization techniques. The research was focused on developing techniques generally not used in the literature in an attempt to overcome some of their limitations.

The first method called the “triangle similarity method” is a geometric interpolation strategy where input data patterns are projected onto a continuous observation space. It is based on the preservation of the geometric similarity between the triangles formed by a sample and two SOM reference vectors on the input space, and by a candidate point and the two corresponding neurons in the observation space. The method calculates the projection by minimizing a cost function that measures distances or errors between triangles, and clearly outperforms other interpolation strategies described in the literature. It can project every input pattern in a nonlinear fashion, it is adequate when the SOM is small, it is robust and can illustrate certain distributions properly that are difficult to analyze with the majority of visualization methods.

Many SOM visualization methods generate monochromatic images that are analyzed individually, providing specific information about the input data distribution. In order to facilitate the analyst’s task of combining information from various methods, a simple superposition method is proposed. It is based on an additive color model where different images are defined using a different color scale and then simply combined by adding their color components. The resulting images are more complete and robust, especially when combining images that provide similar information. The study focuses mainly on combining distance matrices with data histograms.

An alternative to distance matrices, which generate monochromatic images and are the most popular methods for visualizing the data’s cluster structure, is the use of strategies that show different clusters with different colors. One of these strategies is based on neuron contraction models. A new efficient contraction method, the “grouping neuron algorithm”, is presented. Its structure and philosophy is similar to the SOM’s training algorithm, where the concepts are inverted in order to modify the neurons’ positions on the map instead of updating the SOM’s reference vectors. The neurons are attracted to each other depending on the distance between their reference vectors in the input space. Its main advantage is its low computational complexity, which permits the analysis of large maps.

Finally, this dissertation presents an alternative technique based on the explicit visualization on the observable space or map of projected graphs that link neurons whose reference vectors lie close to each other in the input space, like the minimal spanning tree or the “Hebb graph” created with the competitive Hebbian learning principle. The resulting images help analyze the intrinsic dimensionality of the data in different areas of the map and provide a visual and intuitive measure of SOM topology preservation.

# Índice general

Agradecimientos	v
Prefacio	vii
Resumen	ix
Abstract	xi
<b>1. INTRODUCCIÓN</b>	<b>1</b>
1.1. Minería de Datos y Análisis Exploratorio de Datos . . . . .	1
1.2. Mapas Auto-Organizativos . . . . .	3
1.3. Antecedentes y Estado del Arte: Análisis Visual de Mapas Auto-Organizativos . . . . .	5
1.3.1. Técnicas tradicionales . . . . .	5
1.3.2. Técnicas nuevas . . . . .	7
1.3.3. Líneas futuras . . . . .	9
1.4. Objetivo de la Tesis . . . . .	10
1.5. Organización de la Tesis . . . . .	11
<b>2. ANÁLISIS EXPLORATORIO DE DATOS</b>	<b>12</b>
2.1. Definiciones y Conceptos Generales . . . . .	12
2.2. Relación del AED con Técnicas Estadísticas . . . . .	14
2.3. AED y Minería de Datos . . . . .	17
2.4. Visualización de Datos . . . . .	20
2.4.1. Visualización científica . . . . .	20
2.4.2. Aspectos perceptivos y analíticos . . . . .	22
2.5. Métodos de AED . . . . .	28
2.5.1. Introducción a las técnicas gráficas estadísticas tradicionales	29

2.5.2.	Técnicas gráficas estadísticas multidimensionales tradicio- nales . . . . .	31
2.5.3.	Métodos de <i>clustering</i> o cuantización vectorial . . . . .	34
2.5.4.	Métodos de proyección lineales . . . . .	42
2.5.5.	Métodos de proyección no-lineales . . . . .	48
<b>3.</b>	<b>MAPAS AUTO-ORGANIZATIVOS</b>	<b>58</b>
3.1.	Revisión de Redes Neuronales Artificiales . . . . .	58
3.1.1.	Inspiración biológica . . . . .	59
3.1.2.	Desarrollo histórico . . . . .	62
3.1.3.	Fundamentos y características . . . . .	64
3.1.4.	Taxonomía . . . . .	65
3.1.5.	Problemas y aplicaciones . . . . .	66
3.2.	El Mapa Auto-Organizativo Básico . . . . .	67
3.2.1.	Mapas cerebrales . . . . .	67
3.2.2.	Introducción y terminología . . . . .	70
3.2.3.	Arquitectura . . . . .	73
3.2.4.	Algoritmo de entrenamiento . . . . .	77
3.2.5.	Proceso de entrenamiento . . . . .	85
3.2.6.	Ventajas y deficiencias . . . . .	92
3.2.7.	Medidas de bondad . . . . .	96
3.2.8.	Relación entre el EMD y los MAO . . . . .	104
3.2.9.	Visualización, interpretación y uso . . . . .	105
3.3.	Variantes . . . . .	107
3.3.1.	Variantes clásicas de los MAO . . . . .	107
3.3.2.	MAO secuenciales . . . . .	112
3.3.3.	MAO de subespacio adaptativo . . . . .	114
3.3.4.	Cuantización vectorial de aprendizaje . . . . .	115
<b>4.</b>	<b>BASES DE DATOS</b>	<b>116</b>
4.1.	Distribuciones de Simulación . . . . .	116
4.1.1.	Distribuciones bidimensionales . . . . .	117
4.1.2.	Distribuciones tridimensionales . . . . .	121
4.1.3.	Distribuciones multidimensionales . . . . .	125
4.2.	Datos Reales de Habla . . . . .	128

<b>5. MÉTODOS TRADICIONALES DE VISUALIZACIÓN</b>	
<b>DE MAPAS AUTO-ORGANIZATIVOS</b>	<b>133</b>
5.1. Visualización de Distancias . . . . .	134
5.1.1. Rejilla de distancias . . . . .	135
5.1.2. Matrices de distancias . . . . .	137
5.1.3. U-matrix . . . . .	141
5.1.4. Suma de todas las distancias . . . . .	143
5.1.5. Conexiones entre <i>clusters</i> . . . . .	144
5.2. Visualización de Datos . . . . .	144
5.2.1. Unidad de máxima similitud y superficie de respuesta . . .	145
5.2.2. Histograma de datos . . . . .	146
5.2.3. Interpolación básica . . . . .	148
5.2.4. Visualización de trayectorias . . . . .	152
5.2.5. Etiquetado . . . . .	152
5.3. Visualización de Componentes . . . . .	153
5.3.1. Visualización directa . . . . .	155
5.3.2. Planos de componentes . . . . .	155
5.4. Métodos Refinados . . . . .	156
5.4.1. Histogramas de datos suaves . . . . .	159
5.4.2. Contribución de un componente a la estructura de <i>clusters</i>	160
5.4.3. Análisis de bordes . . . . .	161
5.4.4. Interpolación basada en transformaciones afines . . . . .	163
5.5. Proyecciones y Coloración . . . . .	167
5.5.1. <i>Clustering</i> del MAO . . . . .	169
5.5.2. Encadenado de proyecciones mediante códigos de colores .	171
<b>6. NUEVOS MÉTODOS DE VISUALIZACIÓN</b>	
<b>DE MAPAS AUTO-ORGANIZATIVOS</b>	<b>197</b>
6.1. Método de Semejanza de Triángulos . . . . .	197
6.1.1. Descripción del problema . . . . .	198
6.1.2. Solución propuesta . . . . .	200
6.1.3. Resultados experimentales . . . . .	211
6.1.4. Discusión . . . . .	231
6.2. Superposición de Imágenes . . . . .	239
6.2.1. Descripción del problema . . . . .	240
6.2.2. Solución propuesta . . . . .	242

6.2.3.	Resultados experimentales . . . . .	249
6.2.4.	Discusión . . . . .	263
6.3.	Algoritmo de Agrupación de Neuronas . . . . .	264
6.3.1.	Descripción del problema . . . . .	266
6.3.2.	Solución propuesta . . . . .	268
6.3.3.	Resultados experimentales: proyecciones . . . . .	278
6.3.4.	Resultados experimentales: velocidad . . . . .	329
6.3.5.	Discusión . . . . .	348
6.4.	Proyección de Grafos . . . . .	354
6.4.1.	Descripción del problema . . . . .	355
6.4.2.	Solución propuesta . . . . .	358
6.4.3.	Resultados experimentales . . . . .	363
6.4.4.	Discusión . . . . .	380
<b>7.</b>	<b>CONCLUSIONES, APORTACIONES</b>	
	<b>Y LÍNEAS FUTURAS</b>	<b>382</b>
7.1.	Conclusiones y Aportaciones . . . . .	382
7.1.1.	Conclusiones . . . . .	382
7.1.2.	Aportaciones . . . . .	386
7.2.	Líneas Futuras . . . . .	388
<b>A.</b>	<b>SÍMBOLOS, ACRÓNIMOS</b>	
	<b>Y TRADUCCIÓN DE TÉRMINOS</b>	<b>390</b>
A.1.	Símbolos . . . . .	390
A.2.	Acrónimos . . . . .	391
A.2.1.	En español . . . . .	391
A.2.2.	En inglés . . . . .	392
A.3.	Traducción de términos . . . . .	392
	<b>Bibliografía</b>	<b>395</b>

# Capítulo 1

## INTRODUCCIÓN

### 1.1. Minería de Datos y Análisis Exploratorio de Datos

La actual era de la información está caracterizada por un extraordinario crecimiento del volumen de datos, generados y almacenados generalmente en soportes electrónicos, sobre una gran variedad de disciplinas. Una creciente proporción de estos datos es almacenada en forma de bases de datos informáticas, para facilitar su acceso. La disponibilidad de tal cantidad de datos ha generado una demanda de nuevas y potentes herramientas para la extracción de información y conocimiento útil, orientado hacia una determinada aplicación. En un esfuerzo por satisfacer esta necesidad, investigadores han explorado ideas y métodos desarrollados dentro del campo del aprendizaje automático, reconocimiento de patrones, análisis estadístico, visualización de datos, redes neuronales artificiales, etc.

Estos esfuerzos han conducido a la aparición de una nueva área de investigación denominada minería de datos. Ésta está englobada en un marco superior, descubrimiento de conocimiento en bases de datos, cuyo propósito es encontrar nuevo conocimiento en bases de datos de dimensión, tamaño o complejidad demasiado elevados para un análisis manual. Se trata de un proceso interactivo que requiere la intuición y el conocimiento de fondo de expertos, junto con la eficiencia computacional de la tecnología informática moderna [Ves00].

En las primeras fases de un proceso de minería de datos se suele realizar un análisis exploratorio de datos (AED) [Tuk77]. Se trata de una disciplina cuyo objetivo es la formación de visiones globales simplificadas, generalmente de carácter

gráfico, de conjuntos de datos. El AED no comprende una colección de técnicas, sino que debe entenderse como una filosofía sobre cómo debe ser realizado un análisis de datos; sobre cómo examinar un conjunto de datos, qué buscar, cómo buscar y cómo interpretar [FH02]. Tukey utiliza frecuentemente la metáfora del “detective” para referirse al papel del analista en el AED, que es explorar los datos de tantas maneras como sea posible hasta que una “historia” plausible de los datos emerja [Beh97, Yu01]. El enfoque puede resumirse con la siguiente frase: “mirar los datos para ver lo que parecen decirnos” [Tuk77], insistiendo en la idea de que un análisis estadístico debe ser complementado con un análisis visual. Dentro del marco del AED están ubicadas las técnicas clásicas de visualización estadística, además de otras técnicas cuantitativas y gráficas. El peso de los métodos gráficos es debido a su mayor prestación en cuanto a cumplir los propósitos y objetivos del enfoque del AED: permiten explorar y representar los datos en su estructura regular e irregular, y promueven un mayor entendimiento, visión o intuición (*insight*), al aprovechar la capacidad de reconocimiento visual del ser humano [Led02]. De esta manera, se sigue una filosofía en la que “una imagen vale más que mil palabras”, y donde las técnicas deben generar imágenes cuyo valor más importante sea forzarnos a percibir lo que nunca esperábamos ver [Tuk77].

Las técnicas tradicionales de análisis de datos facilitan interpretaciones útiles de éstos y pueden ayudar a comprender los procesos detrás de ellos. Éstas son utilizadas ampliamente para resolver problemas prácticos y están orientadas principalmente hacia la extracción de características estadísticas y cuantitativas. Entre éstas se encuentran técnicas de análisis de regresión, análisis de *clusters*, taxonomías numéricas, análisis multidimensional, métodos estadísticos multivariantes, modelos estocásticos, análisis de series temporales, técnicas de estimación no-lineales, redes neuronales artificiales y otras. No obstante, el conocimiento generado por estas técnicas tiene que ser extraído por analistas humanos, ya que presentan limitaciones inherentes al no caracterizar las dependencias en un nivel conceptual. No pueden producir una descripción cualitativa de las regularidades y determinar sus dependencias en factores que no hayan sido proporcionados explícitamente por los propios datos. Tampoco pueden determinar analogías entre diferentes dominios.

Una alternativa para automatizar el proceso de descubrimiento de conocimiento consiste en utilizar técnicas de aprendizaje automático. Una de las bases



fundamentales de la inteligencia artificial (IA) es la de abordar problemas formalmente intratables mediante la introducción de conocimiento sobre su dominio, codificado en estructuras de datos apropiadas (reglas de producción, marcos, redes semánticas, etc.) cuya implementación deriva en sistemas expertos. El aprendizaje automático se aprovecha de esta idea ya que su objetivo es desarrollar métodos computacionales que implementen varias formas de aprendizaje, en particular, mecanismos capaces de inducir conocimiento a partir de ejemplos de datos. En definitiva, se trata de construir sistemas de aprendizaje que adquieran conceptos de alto nivel y/o estrategias para resolver problemas mediante ejemplos de un modo análogo al aprendizaje humano. Para abordar tareas relacionadas con el análisis automático de datos, un sistema debe ser equipado con una cantidad substancial de conocimiento de fondo, para poder llevar a cabo razonamientos simbólicos sobre la base del conocimiento y de los datos. Actualmente, el aprendizaje automático ha alcanzado un estado de desarrollo en el cual las técnicas y sistemas son capaces de resolver muchos problemas del mundo real. Sin embargo, el proceso de adquisición y codificación de conocimiento en prácticamente cualquier problema real suele ser muy complejo [MBK98].

## 1.2. Mapas Auto-Organizativos

Esta tesis se centra en el análisis visual de mapas auto-organizativos (MAO) [Koh82, Koh98, Koh01]. Se trata de una red neuronal artificial (RNA) competitiva cuyo aprendizaje es no-supervisado. Los MAO sirven para realizar AED y pueden ser considerados dentro de las técnicas de aprendizaje automático.

El algoritmo de los MAO ajusta un conjunto discreto y ordenado de vectores de referencia (centroides, vectores de pesos, vectores del diccionario de bloques) a una distribución de datos de entrada vectoriales definidos en el espacio de los datos (espacio de características, espacio de entrada).

Cada vector de referencia está asociado a una neurona (unidad, nodo, elemento de proceso) en una rejilla regular de éstas. Dicha rejilla discreta, habitualmente denominada mapa, está incluida dentro del espacio de observación (espacio de salida) que suele ser de dimensión menor o igual a tres, por razones de visualización. Los MAO pueden ser vistos como redes elásticas que se adaptan a una distribución de datos de manera ordenada, siguiendo una filosofía similar a

la regresión, de tal forma que posteriormente puede ser posible determinar ciertas características de dicha distribución (*clusters*, distancias relativas entre clases, datos o clases atípicas, etc.) mediante técnicas de inspección visual de ciertas propiedades y medidas asociadas al mapa. Resulta por tanto esencial utilizar técnicas que permitan extraer la máxima información de estas RNA para una posterior visualización [RDG<sup>+</sup>02]. De hecho, el algoritmo de entrenamiento de los MAO fue desarrollado en primer lugar para la visualización de relaciones no-lineales de datos multidimensionales [Koh01].

Los MAO son entrenados con distribuciones de datos de entrada numéricos y pueden ser utilizados como algoritmos de *clustering*, es decir, de cuantización vectorial; y especialmente como algoritmos de proyección vectorial no-lineal, mediante la correspondencia generada por el MAO entre las neuronas y los vectores de referencia asociados a ellas. Esta dualidad está basada en la propiedad de preservación del orden topológico de la proyección, que queda reflejada en un ordenamiento de los vectores de referencia en el mapa.

Aunque la lógica subyacente al modelo de los MAO es generalizable a mecanismos distintos de los neuronales [Koh82], es precisamente este ordenamiento el que relaciona al algoritmo de los MAO con fenómenos de organización en ciertas zonas del cerebro, y por tanto con una idealizada estructura neuronal. En efecto, hay varios tipos de mapas o imágenes de experiencias sensoriales en el cerebro. Las más familiares son las proyecciones retinotópica, somatotópica y tonotópica en áreas sensoriales primarias, además del orden somatotópico de células en la corteza motriz [Koh82].

La preservación de la topología es satisfecha cuando los vectores de referencia de neuronas que se encuentran próximas en el mapa también se localizan próximos en el espacio de los datos. De esta manera, destaca la posibilidad de generar imágenes a partir de dicha proyección que revelen ciertas características inherentes sobre la estructura de la distribución de datos de entrada, generalmente de dimensión elevada, utilizada para entrenar un MAO. De esta manera, el MAO resulta ser una herramienta especialmente indicada para realizar AED, combinando la potencia de un método de aprendizaje computacional con la capacidad natural para el análisis de imágenes del ser humano.

Para una correcta visualización e interpretación de los mapas, éstos deben ser entrenados adecuadamente, para que se ajusten de manera ordenada a la distribución. Esto se comprueba habitualmente con medidas de bondad, que incluyen

medidas de preservación de la topología y otras, como el error de cuantización medio.

Los MAO han resultado ser eficaces a la hora de realizar AED, véanse [Ult93, MJ95, Kas97], y por tanto en tareas de minería de datos y descubrimiento de conocimiento con aplicaciones en, por ejemplo, análisis de textos, datos económicos y sistemas industriales [HKLK97, KK98, SVVH99]. A su vez, han sido utilizados con éxito en una amplia gama de tareas como reconocimiento de patrones, análisis de habla, telecomunicaciones, análisis de imágenes, control de procesos, diagnóstico de fallos, robótica, diseño de circuitos electrónicos, IA, matemáticas, etc. Una extensa recopilación de aplicaciones de los MAO se encuentra en [Koh01].

## 1.3. Antecedentes y Estado del Arte: Análisis Visual de Mapas Auto-Organizativos

### 1.3.1. Técnicas tradicionales

A pesar de la importancia de la tarea de visualización, las técnicas gráficas para analizar MAO no son abundantes en la literatura. El propio Kohonen, a pesar de destacar esta importancia, sólo dedica un breve apartado (*Using Gray Levels to Indicate Clusters in the SOM*) a la visualización en su libro más reciente sobre MAO [Koh01], donde hace referencia al método más popular de análisis: la U-matrix o matriz de distancia unificada [US90, IKKK94], y a variantes descritas en [Ult92, KMJ95, HD95] a las que posteriormente varios autores citan con el nombre de matrices de distancias. Estos métodos tradicionales generan imágenes monocromáticas que muestran información relativa a las distancias entre los vectores de referencia de las neuronas vecinas en el mapa (U-matrix), o a características estadísticas del conjunto de dichas distancias relacionadas con cada neurona (matrices de distancias), como el valor máximo, la mediana, la media, etc. Las imágenes son formadas asociando a cada valor hallado un determinado grado de intensidad luminosa, cuando los MAO son de dimensión menor o igual a dos. El método denominado conexiones entre *clusters* [MR97], muy parecido a la U-matrix muestra diferentes grados de similitud entre neuronas por medio de barras de niveles de gris que conectan neuronas vecinas. Todas estas imágenes son capaces de ilustrar estructuras esenciales de la distribución de datos, sin embargo, pueden presentar deficiencias con determinadas distribuciones al no tener en

cuenta los propios datos para crear las imágenes [RGD<sup>+</sup>02], además de necesitar generalmente un número elevado de neuronas para su correcta aplicación.

En [Ves00] el autor clasifica los métodos de visualización de MAO en tres categorías: métodos de visualización de *clusters*, datos y componentes. Los métodos citados anteriormente están englobados dentro de los métodos de análisis de *clusters*.

Las técnicas relativas a visualizar los datos en el mapa están basadas en la definición del lugar en el espacio de observación donde debe ser ubicada una determinada muestra de la distribución. Existen varias técnicas tradicionales sencillas que incorporan información sobre las muestras de la propia distribución. La más simple consiste en visualizar la unidad de máxima similitud, también denominada “neurona ganadora”, que corresponde a la neurona cuyo vector de referencia es el más próximo a un dato de entrada. Teniendo en cuenta todos los datos es posible generar un histograma de datos, el cual indica la proporción de datos asociada a cada neurona. Por otro lado, si el problema consiste en analizar una secuencia de patrones de entrada, es posible calcular las neuronas ganadoras y conectarlas mediante un grafo dirigido que indique una trayectoria en el mapa, véase [KKS92]. Otro método tradicional de observación basado en los propios datos hace uso de etiquetas que asocia a varias neuronas del mapa. Después de entrenar un MAO datos conocidos *a priori* de varias clases son utilizados para observar las neuronas ganadoras. De esta manera, es posible etiquetar dichas neuronas con nombres de clases conocidas y visualizar la distribución de éstas últimas a lo largo del mapa. El proceso de etiquetado generalmente es manual (al ser supervisado) aunque existen trabajos que lo automatizan. Por ejemplo, el método “labelSOM” [Rau99], que ilustra los nombres de los componentes de los vectores de entrada que más se asemejan a los del vector de referencia que representa a dichos datos de entrada.

La técnica básica de análisis de componentes se denomina planos de componentes. Se trata de visualizar, con niveles de gris y en imágenes separadas, los valores que toman cada una de las variables (componentes) de los vectores de referencia de un MAO entrenado. De esta manera, junto con una técnica de visualización de *clusters*, es posible observar el valor que toma cada variable en cada *cluster*, además de poder establecer dependencias o correlaciones entre las variables. Los planos de componentes pueden ser combinados con otros métodos como la U-matrix para incluir más información en una misma imagen. Varios de

estos métodos de visualización son descritos en [HK97], donde también aparecen diversas estrategias para visualizar los valores de varios componentes de los vectores de referencia, directa y simultáneamente, mediante diagramas de barras o líneas.

#### 1.3.2. Técnicas nuevas

Aproximadamente a partir de 1997 se empezó a desarrollar nuevas técnicas mejoradas de visualización de MAO. Este punto de inflexión puede ser debido al desarrollo de nueva tecnología relativa a interfaces gráficas, librerías gráficas, lenguajes y entornos de programación visuales que facilitan en gran medida el uso de técnicas gráficas; lo que también justificaría la escasez de métodos hasta entonces. Por ejemplo, el uso de colores en gráficas estadísticas fue abandonado durante mucho tiempo debido a la carencia de software y hardware para producir colores [YB95].

A continuación se citan los trabajos recientes más relevantes relacionados con la visualización de MAO, que en general están orientados a visualizar la estructura de los *clusters*. Los nuevos métodos pueden ser clasificados en dos categorías: estrategias para refinar los métodos anteriores, por ejemplo, métodos híbridos para combinar información de distancias, de los propios datos y de sus variables; y nuevos métodos de proyección y coloración.

En cuanto a la visualización de datos, en [PRM02] se emplea una técnica difusa que estima la densidad de probabilidad de la distribución de datos en el mapa. Utiliza histogramas de datos suaves (continuos) para ilustrar los *clusters* con niveles de gris de manera similar al método “generative topographic mapping” [BSW98].

Otra forma de visualizar los datos, generalmente no comentada en la literatura, es proyectar los datos de la distribución sobre el propio mapa, siguiendo una filosofía de interpolación que tenga en cuenta las posiciones de las neuronas en el mapa y los vectores de referencia de éstas. En [GR93, GR95a] se propone un método de interpolación basado en coordenadas afines de un sistema local. Éste presenta varios problemas que el mismo autor comenta, y propone un método iterativo para estimar varios parámetros para mejorar su rendimiento [GR95b].

En [KNK98] se describe un método que puede ser englobado también dentro de las técnicas de análisis de componentes, en el que se estudia la contribución

explícita de las variables de la distribución a la estructura del MAO, y por consiguiente a la de los *clusters*. El método genera imágenes monocromáticas de forma similar al U-matrix, pero considerando solamente un componente.

Por otro lado, varios nuevos métodos han sido introducidos que combinan la información de las distancias representadas en la matriz de la U-matrix con información acerca de los datos. En [KNK00, NTK<sup>+</sup>01] el método propuesto genera imágenes monocromáticas en las que se mejora el análisis de los bordes entre los *clusters*. El método tiene en cuenta la distancia entre los vectores de referencia de las neuronas y la media o centroide de los datos de la distribución asociados a dicha neurona (pertenecientes a la clase representada por el vector de referencia).

Varios autores presentan métodos alternativos basados en proyecciones no-lineales de los vectores de referencia al espacio de salida, real y generalmente bidimensional, en el que se define el MAO para analizar la estructura de los *clusters*. El primer método descrito en [MR97], denominado coordenadas adaptativas, está basado en el propio algoritmo de entrenamiento de los MAO, y simula los desplazamientos de los vectores de referencia durante el proceso de aprendizaje, traduciéndolos a desplazamientos relativos de las propias neuronas en un plano real. Este método resulta interesante no sólo porque permite analizar la estructura de los *clusters*, sino porque ofrece una visión (en una secuencia dinámica de imágenes) sobre el proceso de entrenamiento y la adaptación del MAO a la distribución de datos. Por otro lado, en [Him00] se describe un modelo de contracción en el que las neuronas en el mapa se aproximan entre sí teniendo en cuenta la similitud entre sus vectores de referencia. Esto produce una segunda proyección que genera visualizaciones en forma de árboles jerárquicos (dendogramas), donde es posible ver la estructura de los *clusters*.

De esta manera, aparece un nuevo concepto en la literatura: el encadenado de dos proyecciones de los vectores de referencia al espacio de observación (véase [KK98]). El encadenado es una simple asociación entre las dos proyecciones y su utilidad radica en asignar a una proyección ciertas características de la otra. En el caso que nos ocupa, el encadenado suele ser llevado a cabo mediante el uso de un código de colores, en el que se realiza una asignación de colores a los vectores de una primera proyección (por ejemplo, en función de un código lineal), y posteriormente la segunda proyección es coloreada en función de la asociación entre ambas. El método de Himberg [Him00] aprovecha de esta idea

asignando colores mediante un código lineal a la proyección que genera, para posteriormente colorear las neuronas de la proyección original regular del MAO. De hecho, una reciente línea de investigación está desarrollando nuevos métodos más eficaces para analizar la información contenida en los MAO. Éstos están basados en el encadenado y en los códigos de colores para generar imágenes más fáciles de interpretar, mejorando notablemente el análisis del MAO, y por tanto el análisis exploratorio de las distribuciones. Las imágenes resultantes muestran cada *cluster* con un color diferente y las diferencias entre dichos colores indican la relación entre los propios *clusters*. En [Him98, KK98] se utilizan códigos de colores lineales para encadenar la proyección de un MAO con otras técnicas de proyección lineal (diagramas de dispersión) o no-lineal como el escalado multidimensional, por ejemplo, la conocida proyección de Sammon [Sam69]. En [KVK99, KVK00] se utiliza un código de colores no-lineal, basado en el espacio de color CIELab, de manera que las diferencias entre los vectores de referencia vecinos queden reflejadas en diferencias perceptivas en la imagen del nuevo mapa coloreado. Una recopilación de varias de estas técnicas se encuentra en [Ves99, VA00].

### 1.3.3. Líneas futuras

Actualmente, la mayoría de trabajos que utilizan MAO para realizar AED de distribuciones de dimensión elevada, incluida esta tesis, hacen uso de mapas bi-dimensionales por razones de visualización en un soporte bidimensional (pantalla de un monitor, papel, etc.). Sin embargo, teóricamente la dimensión del MAO debe coincidir con la dimensión intrínseca de los datos para que la proyección generada por el MAO preserve la topología, en la mayor medida posible, y éste se pueda adaptar a dicha distribución para representarla lo más fielmente posible. En este sentido, cabe destacar la necesidad de nuevos métodos para analizar MAO de dimensión tres o más, en los soportes que nos ofrece la tecnología actual. Aunque en [Kiv98] se analiza un MAO tridimensional mediante la visualización de varias U-matrices y planos de componentes simultáneamente, los trabajos en esta dirección son muy escasos.

Por otro lado, no hay que olvidarse de aprovechar al máximo las nuevas tecnologías del momento, como hizo Santiago Ramón y Cajal (1852–1934), premio Nobel de medicina y fisiología en 1906, para muchos “el científico más grande que ha tenido España” [La202]. Ramón y Cajal utilizó los microscopios, técnicas fotográficas y técnicas de preparación de muestras más vanguardistas (estereoscopia,

coloración, impregnación de muestras con plata, uso de tejido embrionario, etc.) para descubrir, por ejemplo, la individualidad de las neuronas [Ram77, La202]. Nosotros debemos hacer uso de los avances en visión tridimensional, que están empezando a proporcionar una tecnología que nos va a permitir analizar MAO tridimensionales, más generales y potentes a la hora de analizar distribuciones de dimensión elevada. Varias de las técnicas comentadas y en particular las nuevas propuestas en esta tesis podrán beneficiarse de esta tecnología, la cual seguramente provocará una nueva generación de métodos para el análisis visual de MAO, además de promover nuevas técnicas de visualización estadística y análisis multivariante.

## **1.4. Objetivo de la Tesis**

El objetivo principal de esta tesis es desarrollar nuevas técnicas de visualización para analizar MAO; tanto desde el punto de vista del análisis exploratorio de la distribución de datos con la que se ha entrenado un MAO, como desde el punto de vista de comprender el proceso de adaptación del MAO, que puede servirnos como una medida de bondad visual. Por otro lado, un segundo objetivo consiste en llevar a cabo una recopilación de las técnicas más relevantes aparecidas en la bibliografía sobre el AED (visual) mediante MAO.

La motivación para desarrollar nuevas técnicas de visualización surge por:

- La relativa carencia de métodos destinados a la importante tarea de visualización.
- La necesidad de analizar MAO con varios métodos diferentes.
- La necesidad de mejorar varios métodos descritos en la literatura.
- La posibilidad de innovar desarrollando nuevas estrategias de visualización.

De esta manera, se ha hecho hincapié en desarrollar técnicas generalmente no utilizadas con anterioridad en un intento por superar limitaciones de varios métodos descritos en la literatura. En concreto, se propone abordar las siguientes estrategias:

- Interpolación o proyección eficaz y robusta de los datos sobre un mapa o espacio de observación continuo.



- Superposición de imágenes monocromáticas para crear imágenes coloreadas más completas y robustas.
- Contracción eficiente de neuronas basada en algoritmos auto-organizativos.
- Visualización explícita sobre el mapa de grafos que conectan vectores de referencia próximos en el espacio de los datos.

## 1.5. Organización de la Tesis

Los primeros capítulos de esta tesis proporcionan una base esencial y un resumen del trabajo relacionado con el AED y los MAO.

2. Revisión del campo del AED. Se introduce su filosofía y relación con otros campos, y se lleva a cabo una recopilación de métodos de análisis exploratorio multivariante, haciendo hincapié en métodos gráficos.
3. Revisión de MAO. El capítulo repasa brevemente conceptos generales de RNA para posteriormente abordar los MAO en detalle (algoritmos, aspectos relevantes para su empleo correcto, variantes, etc.).

El resto de la tesis está relacionada con los nuevos métodos de visualización de MAO fruto de la labor de investigación llevada a cabo:

4. Introducción a los conjuntos de datos utilizadas en los diversos ejemplos y experimentos.
5. Recopilación y análisis de las técnicas de visualización de MAO más relevantes de la literatura.
6. Nuevas técnicas de visualización de MAO. Este capítulo representa el trabajo innovador llevado a cabo. Describe varios nuevos métodos de visualización de MAO que complementan o sustituyen a los métodos anteriores, exponiendo su filosofía, diseño, implementación y resultados.
7. El último capítulo describe las principales conclusiones, aportaciones y líneas futuras de esta tesis.

Finalmente, el Apéndice recoge varias listas de símbolos, acrónimos y términos traducidos del inglés al español utilizados a lo largo de la tesis.

## Capítulo 2

# ANÁLISIS EXPLORATORIO DE DATOS

Dada la función principal de los MAO, que es analizar distribuciones multivariantes, éstos resultan ser herramientas muy indicadas para llevar a cabo análisis exploratorio de datos (AED). Este capítulo presenta varias definiciones y conceptos generales sobre el AED, su relación con los métodos estadísticos clásicos de análisis y visualización, su papel dentro del marco de la minería de datos, una revisión de la teoría relacionada con la visualización de datos, y una amplia colección de técnicas que pueden servir para realizar dicho AED.

### 2.1. Definiciones y Conceptos Generales

El AED [Tuk77] se refiere a una tradición específica de análisis de datos cuya raíz se encuentra en el trabajo pionero John Tukey y sus colegas a mediados de los años sesenta. Puede ser definido como una disciplina cuyo objetivo es la formación de visiones globales simplificadas, generalmente de carácter gráfico, de conjuntos de datos. Es una actitud, enfoque o filosofía para el análisis de datos en la que se persigue [FH02]:

- Maximizar el conocimiento sobre un conjunto de datos.
- Desvelar su estructura fundamental.
- Extraer variables importantes.
- Detectar anomalías o datos atípicos.

- Probar supuestos.
- Desarrollar modelos parsimoniosos.
- Determinar grupos de factores óptimos.

El AED está relacionado con los métodos gráficos estadísticos, los cuales reúnen un conjunto de técnicas centradas en visualizar un determinado aspecto característico de los datos. Sin embargo, éste es más amplio y debe ser entendido, no como un conjunto de técnicas, sino como una filosofía relativa al modo de inspeccionar un conjunto de datos, definir qué se busca, cómo buscar y cómo interpretar. La idea principal es proporcionar al analista un apoyo visual para explorar, intentando que los datos desvelen sus secretos estructurales, y descubrir, a veces de forma inesperada, nueva información sobre ellos [FH02]. Desde un punto de vista cognitivo, se sigue una filosofía en la que “una imagen vale más que mil palabras”, ya que resulta más sencillo detectar patrones en datos inspeccionando una imagen que examinando una lista numérica [Yu01]. En combinación con nuestras capacidades naturales de clasificación y discriminación, las imágenes o gráficas nos proporcionan una potencia extraordinaria para examinar datos.

En un típico proceso de AED, muchas variables son consideradas y comparadas, utilizando diferentes técnicas en la búsqueda de patrones sistemáticos [Sta03a]. Tukey utiliza frecuentemente la metáfora del “detective” para referirse al papel del analista [Tuk77], que es explorar los datos de tantas maneras como sea posible hasta que una “historia” plausible de los datos emerja [Beh97, Yu01]. El enfoque puede resumirse con la siguiente frase: “mirar los datos para ver lo que parecen decirnos”, y donde “las técnicas deben generar imágenes cuyo valor más importante sea forzarnos a percibir lo que nunca esperábamos ver” [Tuk77]. Por otro lado, Tukey insiste en la idea de que un análisis estadístico (análisis confirmatorio de datos) debe ser complementado con un análisis visual (AED).

Por otra parte, Velleman y Hoaglin [VH81], sugieren los siguientes elementos generales del AED, también resumidos y desarrollados en [Beh97, Yu01]:

- Visualización de Datos. Se trata de utilizar técnicas gráficas para profundizar en el conocimiento de los datos. Este punto es desarrollado en la Sección 2.4.
- Análisis de residuales. Un examen de la desviación los datos con respecto a un modelo propuesto, para comprobar si éste es adecuado.

- Transformación y re-expresión de los datos. Los tres objetivos principales a la hora de transformar los datos son: normalizar la distribución, ya que datos no-normales violan los supuestos de tests paramétricos; estabilizar las varianzas, datos con varianzas desiguales ocasionan diversos problemas a los métodos de visualización, clasificación y tests paramétricos; y linealización de la tendencia, para permitir análisis como el de regresión, el cual requiere el supuesto de linealidad.
- Robustez de los procedimientos. Por ejemplo, los tests paramétricos están basados en la estimación de la media, que es sensible a datos atípicos. El AED utiliza estimadores robustos o “resistentes” como la mediana u otros.

## 2.2. Relación del AED con Técnicas Estadísticas

Las técnicas tradicionales de análisis de datos facilitan interpretaciones útiles de éstos y pueden ayudar a comprender los procesos detrás de ellos. Éstas son utilizadas extensamente para resolver problemas prácticos y están orientadas principalmente hacia la extracción de características estadísticas y cuantitativas. Entre éstas se encuentran técnicas de análisis de regresión, análisis de *clusters*, taxonomías numéricas, análisis multidimensional, métodos estadísticos multivariantes, modelos estocásticos, análisis de series temporales, técnicas de estimación no-lineales, RNA y otras.

Dentro del marco del AED se pueden ubicar las técnicas clásicas de visualización estadística, además de otras técnicas cuantitativas y gráficas. La visualización estadística se ocupa de los fundamentos, investigación, desarrollo y aplicación de métodos gráficos para el análisis de datos. Ello incluye aspectos estadísticos, cognitivos, informáticos y de diseño entre otros [Led02]. La visualización estadística puede ser incluida también dentro del campo más general de la visualización científica, véase la Sección 2.4.1.

La diferencia principal entre el AED y los métodos clásicos (estadísticos) de análisis de datos radica en el orden en el que se lleva a cabo dicho análisis de datos y la creación de un modelo de éstos, véase la Figura 2.1. El AED primero analiza los datos para posteriormente crear un modelo, mientras que las técnicas clásicas primero imponen un modelo (lineal, normal, etc.) para posteriormente analizar

## 2.2. RELACIÓN DEL AED CON TÉCNICAS ESTADÍSTICAS

---

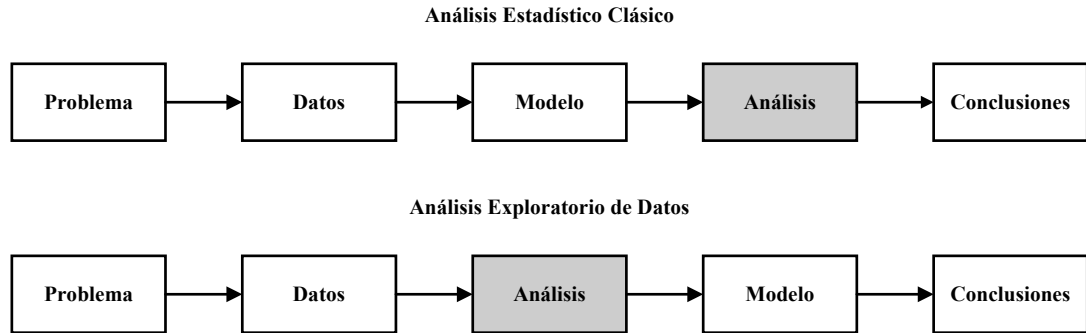


Figura 2.1: Secuencia de etapas del análisis clásico y exploratorio.

dichos datos. Por tanto, en el análisis clásico la recolección de datos es seguida de la imposición de un modelo, el análisis, la estimación y las pruebas posteriores que se centran en los parámetros del modelo. En el AED, después de la recolección de datos se realiza un análisis con el propósito de inferir un modelo apropiado [FH02]. De esta manera, en lugar de verificar hipótesis *a priori* sobre las relaciones entre las variables, el AED es aplicado para identificar relaciones sistemáticas entre las variables, teniendo en cuenta que no hay información completa *a priori* sobre la naturaleza de dichas relaciones [Sta03a]. El AED hace énfasis en buscar estructuras inesperadas y desarrollar descripciones “ricas” mediante resúmenes gráficos, estadísticos robustos e indicadores de ajuste de modelos. En los métodos tradicionales de análisis confirmatorio se asume una estructura subyacente de los datos y se procede con la inferencia sobre la base de supuestos como normalidad, homogeneidad e independencia [YB95].

Las diferencias más significativas entre el AED y el análisis clásico de datos son las siguientes [FH02]:

- **Modelo.** El AED no impone un modelo determinista ni probabilístico sobre los datos, permitiendo que los propios datos sugieran modelos admisibles que se ajusten mejor a su distribución.
- **Enfoque.** En el análisis clásico el enfoque se centra en la estimación de los parámetros del modelo y en la predicción de valores a partir del modelo. El AED se centra en la estructura, datos atípicos y modelos sugeridos por los propios datos. Cabe destacar que muchos métodos clásicos de análisis multivariante están basados únicamente en la estructura de correlaciones del conjunto de datos, por lo que son inadecuados para tratar *clusters*, datos

atípicos u otras estructuras. En estas situaciones es conveniente utilizar los métodos de AED, en los cuales la “estructura” puede ser definida prácticamente como uno desee [Nas92].

- **Técnicas.** Las técnicas clásicas son de naturaleza cuantitativa. Éstas incluyen técnicas de análisis de la varianza, pruebas T, pruebas chi-cuadrado y pruebas F. Las técnicas empleadas para realizar AED son generalmente gráficas, incluyendo técnicas de visualización estadística clásicas (histogramas, diagramas de dispersión, diagramas de cajas, matrices de dispersión, etc.), y otras técnicas de visualización multivariante (técnicas de clasificación o *clustering*, técnicas basadas en proyecciones, etc.). Una amplia descripción de estas técnicas es presentada en la Sección 2.5.
- **Rigor.** Las técnicas clásicas dan un soporte probabilístico a la ciencia e ingeniería, son rigurosas, formales y objetivas. Las gráficas estadísticas deben comunicar ideas sobre los datos con precisión, claridad y eficiencia. Se deben evitar imágenes que distorsionen los datos y dificulten su comprensión y comparación [Fri03]. Por el contrario, las técnicas de AED son subjetivas y su interpretación puede variar de un analista a otro, aunque analistas experimentados suelen llegar a las mismas conclusiones. La carencia de rigor es contrarrestada con las capacidades de sugerencia, indicación y visión sobre el modelo adecuado que debe ser empleado.
- **Tratamiento de Datos.** Las técnicas clásicas resumen los datos en un número reducido de “estimaciones”. Aunque estos valores se centran en aspectos importantes de los datos, en este “filtrado” se produce una pérdida de información. El enfoque del AED puede utilizar y mostrar todos los datos de la distribución sin pérdida de información.
- **Supuestos.** La validez de las conclusiones científicas está intrínsecamente ligada a la validez de los supuestos subyacentes al realizar un análisis clásico. Si éstos son desconocidos o no han sido probados se pone en duda la validez de las conclusiones. Por el contrario, las técnicas de AED hacen muy pocos o ningún supuesto y presentan todos los datos tal y como son.

## 2.3. AED y Minería de Datos

La minería de datos es un proceso analítico diseñado para explorar datos, generalmente gran cantidad de ellos, en busca de patrones consistentes, tendencias y/o relaciones sistemáticas entre variables, para después validar lo descubierto aplicando los patrones detectados a nuevos conjuntos de datos [Sta03a]. La minería de datos puede ser considerada un híbrido de la estadística, la IA y la investigación de bases de datos [Pre96]. Su objetivo es encontrar respuestas a ciertas preguntas a partir de un conjunto de datos, lo cual es un tema de investigación con profundas raíces en estadística. Sin embargo, en la última década se ha establecido como un área independiente de investigación. De hecho, el término minería de datos tenía un significado negativo en el pasado, al referirse al peligro de encontrar patrones incluso cuando éstos no existen [Ves00].

La minería de datos puede ser englobada en el marco superior del descubrimiento de conocimiento en bases de datos, cuyo propósito es encontrar nuevo conocimiento en bases de datos de dimensión, tamaño o complejidad demasiado elevados para un análisis manual. Se trata de un proceso interactivo que puede utilizar técnicas de AED y que requiere la intuición y el conocimiento de fondo de expertos, junto con la eficiencia computacional de la tecnología informática moderna [Ves00]. Este campo hace énfasis en todo el proceso interactivo de descubrimiento de conocimiento, nuevos patrones o estructuras en los datos. El proceso consta de varios pasos, desde especificar los objetivos hasta evaluar los resultados, y posiblemente reformular los objetivos en función de los resultados. La minería de datos es una de estas etapas, en la que en función de sus objetivos, caben herramientas de otras disciplinas como el reconocimiento de patrones, aprendizaje automático, análisis multivariante y el citado AED, entre otras [Kas97].

Dada su reciente aparición y desarrollo, existen varias definiciones relativas al término minería de datos dependiendo de su propósito. Si éste es el descubrimiento de conocimiento, dicho conocimiento debe ser representado simbólicamente en un lenguaje formal, debe ser comprensible para seres humanos y debe servir como apoyo a los sistemas basados en conocimiento. En este sentido, una cuestión fundamental es la transición de los datos al conocimiento o “conversión de conocimiento”. Ésta es omitida en muchas aplicaciones comerciales de minería de datos, las cuales utilizan herramientas estadísticas con “adornadas” interfaces visuales [Ult99].

Aunque siempre existe una medida de incertidumbre en los resultados de algoritmos de descubrimiento de conocimiento, éstos parecen adecuados en un marco de AED [Hea96]. Recíprocamente, las técnicas de descubrimiento de conocimiento pueden mejorar el análisis de datos multidimensionales. En este sentido se situarían entre la herramienta de visualización y el conjunto de datos, actuando de filtro para remarcar o comprimir regiones de interés de los datos. Varias de éstas técnicas pueden ser aplicadas conjuntamente y pueden mejorar el proceso de visualización de las siguientes maneras [Hea96]:

- Reduciendo la cantidad de datos a visualizar, mediante agrupación o clasificación, ignorando grupos que no contribuyen a la relación investigada.
- Reduciendo la cantidad de atributos de cada elemento, ignorando atributos independientes de la relación investigada, descubriendo clasificaciones que reemplacen dimensiones múltiples de los datos o eliminando atributos redundantes.
- Mostrando la estructura del conjunto de datos.
- Comprimiendo datos variantes en el tiempo en el eje temporal, lo cual reduce la cantidad de datos a visualizar.
- Encontrando y visualizando tendencias o relaciones previamente desconocidas entre los elementos del conjunto de datos.

Por otro lado, el propósito principal de la minería de datos puede ser la predicción, en cuyo caso el proceso puede ser dividido en las tres siguientes fases [Sta03a]:

1. Exploración. Esta etapa suele comenzar por una preparación de los datos que puede involucrar una “limpieza”, transformación y selección de subconjuntos de los datos, además de una selección preliminar de las variables a tratar. Posteriormente, dependiendo de la naturaleza del problema el proceso puede involucrar desde técnicas estadísticas sencillas hasta análisis exploratorios más sofisticados utilizando la gran variedad de técnicas gráficas y estadísticas asociadas al AED.
2. Construcción de modelos y validación. En esta etapa se trata de considerar varios modelos y elegir el mejor en función de su comportamiento predictivo.



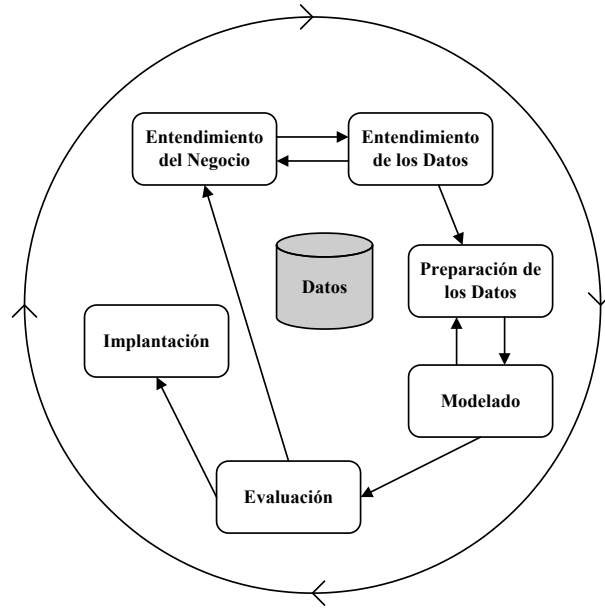


Figura 2.2: *CRISP-DM*: modelo de proceso estándar para la minería de datos.

3. Implantación. En la que el mejor modelo de la etapa anterior es utilizado y aplicado a nuevos datos con el objetivo de generar nuevas predicciones o estimaciones.

Para facilitar el proceso de la minería de datos varios modelos han sido propuestos para formalizarlo. Destaca el modelo *CRoss Industry Standard Process model for Data Mining (CRISP-DM)* [CCK<sup>+</sup>99], creado a mediados de los noventa por un consorcio europeo de compañías, para servir como estándar para el proceso. Se trata de definir la minería de datos como un proceso cíclico, ilustrado en la Figura 2.2, que consta de las siguientes etapas: entendimiento del negocio, entendimiento de los datos, preparación de los datos, modelado, evaluación e implantación. Los bucles son implementados ya que el proceso puede producir nueva información en cada iteración que vaya mejorándolo [Ves00].

La minería de datos sigue estando basada en los principios conceptuales de la estadística, incluyendo el AED y el modelado, y comparte con ellos algunos componentes de sus técnicas específicas y enfoques generales. Como se ha visto anteriormente, en las primeras fases de un proceso de minería de datos suele llevarse a cabo un AED. Sin embargo, existe una importante diferencia general en el propósito y enfoque entre el AED y la minería de datos. Esta última está más

orientada hacia aplicaciones que hacia la naturaleza subyacente de los fenómenos. En otras palabras, la minería de datos se preocupa menos en identificar las relaciones específicas entre las variables, centrándose en producir soluciones que puedan generar predicciones útiles. De esta manera, acepta enfoques de tipo “caja negra”, como ciertos tipos de RNA que generan predicciones válidas aunque son incapaces de describir la naturaleza específica de las interrelaciones entre las variables sobre las cuales se basan las predicciones [Sta03a].

## 2.4. Visualización de Datos

Las técnicas modernas de AED hacen uso de imágenes generadas por métodos computacionales. Para que éstas resulten útiles, deben tenerse en cuenta ciertos aspectos teóricos sobre visualización de datos, los cuales son introducidos brevemente en este apartado.

### 2.4.1. Visualización científica

La visualización científica puede ser definida como “el proceso de explorar o mostrar datos de manera que se construya una analogía con el mundo físico para servir al usuario de conocimiento y aprendizaje” [YB95]. El término fue introducido en un informe de un tribunal de la *National Science Foundation* en 1987 sobre la aplicación de técnicas informáticas a problemas de análisis de datos [MDB87]. El tribunal predijo que las técnicas de visualización científica podrían ser útiles para comprender la evolución de modelos computacionales, para ajustar ciclos de diseño, para integrar herramientas de hardware y software, y para estandarizar interfaces gráficas de usuario. La capacidad para visualizar simulaciones y computaciones complejas es absolutamente esencial para asegurar la integridad del análisis, profundizar en su comprensión, y poder divulgarlo. Por otro lado, como alternativa a los números las imágenes resultan ser una posibilidad cognitiva y una realidad técnica, por lo que se afirmó que de manera significativa más complejidad puede ser comprendida mediante técnicas de visualización en computación científica que mediante técnicas clásicas [MDB87]. El tribunal hizo énfasis en una serie de objetivos de investigación, en concreto, la visualización [Hea96]:

- Debe ser un campo interdisciplinario en el se deben combinar resultados de

investigación en informática, gráficas por computador, tratamiento de señal y de imágenes, psicología, artes visuales, diseño asistido por computador, interfaces de usuario, etc.

- Debe referirse a un modo inteligente de mostrar los datos y a un modo inteligente de gestión del conjunto de datos subyacente.
- No debe ser complicada para que sea útil.
- Debe ser llevada a cabo de forma interactiva mientras los datos son generados.

Para el tribunal, “la visualización ofrece un método para ver lo no visto, enriquece el proceso de descubrimiento científico y fomenta profundas e inesperadas visiones” [MDB87]. Dado que más del 50 % de las neuronas del cerebro están asociadas a la visión, McCormick y sus colegas sugirieron que la visión humana sería el método más eficaz para tratar grandes volúmenes de datos [Bre03]. Desde entonces, fabricantes de hardware y software han invertido recursos para mejorar las capacidades de los gráficos por computador. La incorporación de la informática permite además interacción por parte del usuario, visualización de resultados en tiempo real y generación de gráficos dinámicos. Esto confiere a las técnicas mayor capacidad para transmitir información y mayores prestaciones en situación de análisis exploratorio multivariante [Led02]. Estos desarrollos pueden ser vistos como una extensión lógica del trabajo de Tukey [Yu95].

Miembros de un tribunal similar al anterior notaron que la demanda de una visualización basada en computadoras surgió de la necesidad de analizar conjuntos de datos cada vez más extensos y complejos [WF88]. La visualización científica ha crecido notablemente en los últimos años como resultado directo de la cantidad enorme de datos que está siendo generada. Nuevas técnicas de visualización necesitan ser desarrolladas para hacer frente a este torrente de información si los usuarios esperan analizar incluso una pequeña fracción de sus repositorios de datos [Hea96].

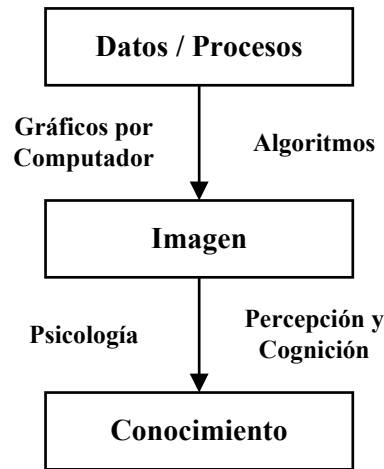


Figura 2.3: Proceso de visualización.

## 2.4.2. Aspectos perceptivos y analíticos

### 2.4.2.1. Visualización = algoritmos + percepción

Una verdadera ciencia de visualización de datos requiere tanto una teoría de percepción como de técnicas gráficas por computador. Los diseñadores de herramientas de visualización han prestado relativamente poca atención a aspectos perceptivos, a pesar de que conocimiento sobre la percepción y fisiología visual humana puede conducir a visualizaciones más efectivas [Gre98].

La transformación de números a conocimiento requiere dos fases, véase la Figura 2.3. La primera emplea un algoritmo para generar una imagen, mientras que la segunda transforma el contenido de la imagen a conocimiento basándose en tareas perceptivas y cognitivas. La literatura de investigación sobre visualización y gráficos suele estar escrita por informáticos quienes describen nuevas técnicas para generar imágenes, muchas veces ignorando la segunda fase al no emplear modelos explícitos de percepción, basándose únicamente en su intuición e introspección [Gre98]. La mayoría de métodos de visualización de AED, en efecto, no tiene en cuenta dichos aspectos de percepción y cognición. En esta tesis, los algoritmos propuestos constan generalmente de dos fases: una analítica o algorítmica, con cierta independencia de los factores de percepción; y otra de visualización, en la que se ha procurado facilitar la extracción de conocimiento en función de dichos factores, especialmente a la hora de emplear colores y tonos de gris.

Investigadores en psicología y visión han trabajado para explicar como el sistema visual humano analiza imágenes. La codificación visual de información gráfica involucra tareas gráficas de percepción (procesado “preatento”) y cognición (procesado “atento”). La diferencia entre estas dos radica en el nivel mental en el que ocurren. La percepción puede ser llevada a cabo mediante un simple vistazo, donde ciertas características son detectadas inmediatamente por el sistema visual (juicio del ángulo, área, matiz del color, saturación del color, densidad, distancia, pendiente, etc.). Por otro lado, las tareas cognitivas requieren el estudio de la gráfica un cierto periodo de tiempo e involucran cálculos mentales. Aunque fuera del alcance de esta tesis, se recomienda el estudio de la teoría de imagen de Bertin [Ber83], quizás el trabajo más exhaustivo sobre teoría perceptiva en la literatura sobre visualización [Gre98]. El interés de Bertin se centra en crear buenas visualizaciones de datos multidimensionales. Básicamente, su teoría de percepción gráfica consiste en la descomposición de una imagen en un conjunto de componentes primitivos y la especificación de una serie de procedimientos que combinan dichos componentes para crear buenas visualizaciones [Gre98]. Sin embargo, con respecto al AED, las teorías sobre formatos gráficos suelen centrarse en la presentación gráfica con el fin de comunicar un resultado del análisis, y no en la visualización de datos como un proceso de descubrimiento [Yu95].

### 2.4.2.2. Visualización multidimensional

Las características perceptivas pueden ser utilizadas para realzar características importantes en las imágenes [Hea96]. Bajo el contexto de la visualización multidimensional, experimentos en psicología han utilizado estas características para llevar a cabo las siguientes tareas relacionadas con un rápido y preciso AED [Hea96]:

- Detección de objetivo. Detección de la presencia o ausencia de un elemento objetivo en la imagen.
- Detección de bordes. Identificación de las fronteras entre grupos de elementos o *clusters*, donde los elementos de un grupo comparten propiedades en común.
- Rastreo de regiones. Localización de grupos de elementos con una propiedad en común mientras se mueven a través del tiempo y/o del espacio.

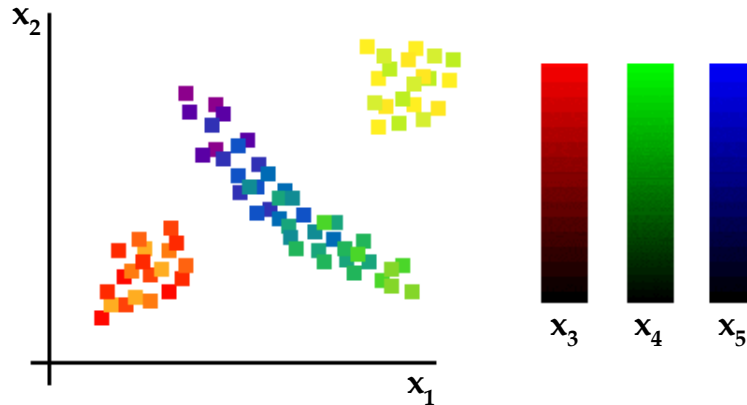


Figura 2.4: Visualización de datos 5–dimensionales utilizando dos dimensiones espaciales y tres basadas en códigos de colores.

- Estimación y Cuantificación. Estimación del número o porcentaje de elementos visualizados con una determinada característica.

De esta manera, es posible identificar áreas que puedan beneficiarse de una investigación posterior más detallada.

Con respecto a la visualización multidimensional, el ser humano puede percibir valores de datos de dos variables representadas en ejes espaciales, sin embargo, tiene problemas con dimensiones más elevadas. Dada la bidimensionalidad de las imágenes representadas en los medios que ofrece la tecnología actual, las técnicas de visualización multidimensional deben representar de alguna manera dimensiones superiores mediante características no espaciales. Por ejemplo (véase la Figura 2.4), datos 5–dimensionales pueden ser representados de la siguiente manera: las dos primeras componentes en un plano y las otras tres mediante las componentes del modelo de color RGB: rojo, verde y azul. Parece claro que el ser humano percibe datos codificados en dimensiones espaciales mucho más fácilmente que en no-espaciales. Esto puede ser debido a las dos dimensiones espaciales representadas en la imagen de la retina, e incluso a un tratamiento especial de éstas en el cerebro, en perjuicio de las no espaciales. La tercera dimensión de profundidad espacial se puede añadir aunque su utilidad parece limitada [Gre98].

Por otro lado, varios métodos de visualización multidimensional introducen un componente de subjetividad. En estos casos, es necesario que varias personas colaboren a la hora de realizar los análisis. Por otro lado, para medir la efectividad y rendimiento de un método determinado de visualización, éste debe ser analizado con numerosas distribuciones de datos y por varios analistas.

Uno de los problemas que puede afectar a la visualización y que está relacionado en general con los conjuntos de datos multidimensionales es la denominada “maldición de la dimensionalidad”. Se trata del hecho de que la mayor parte del espacio multidimensional está vacío, incluso cuando la cardinalidad de los datos es considerable. Algunas técnicas estadísticas presentan limitaciones causadas por esta demanda de datos, que crece exponencialmente con la dimensión del espacio de los datos. Por fortuna, existen técnicas de proyección capaces de evitar este problema [SSS88].

### 2.4.2.3. Color

Aunque existen diversas características no-espaciales que pueden ser utilizadas para proporcionar más información a las imágenes, como la orientación o la textura de los elementos mostrados, este apartado se centra en el color, dados los trabajos presentados en esta tesis.

El uso del color en gráficos estadísticos ha sido poco frecuente hasta hace poco debido a la carencia de software y hardware orientado a la producción de éste [YB95]. Sin embargo, la tecnología reciente facilita el uso del color y mejora varios métodos de AED, ya que éste proporciona mayor información al usuario. De esta manera, el color es una característica visual a destacar que cada vez se hace más popular. En las últimas décadas, ha habido un incremento del uso del color, el cual ya no es considerado como una mera extensión de una escala de grises, sino que es visto como un elemento clave en los sistemas de tratamiento de imagen y vídeo. El color juega un papel cada vez más importante en sistemas de reconocimiento de patrones y segmentación de imágenes. Los informáticos usan modelos de color, generalmente tridimensionales, para describir los diferentes colores. Entre ellos destacan el RGB (para monitores de color), CMY (para impresoras de color), YIQ (para señal de televisión por color), HSV, CIEXYZ, CIELUV, CIELab y Munsell, véanse [Rus95, HE96, Cas96, Fai98].

Desafortunadamente, la elección de los colores para visualización de datos es complicada, dado que varios factores diferentes pueden afectar las interacciones que ocurren entre los colores. Por otro lado, el uso de demasiados colores puede complicar la interpretación y comprensión de las imágenes y por tanto de los datos. En [Hea96] se describe un extenso estudio y recopilación sobre la aplicación de colores para analizar distribuciones de datos. La cuestión principal a la hora de emplear color es seleccionar colores efectivos que proporcionen una buena

discriminación entre los datos en la tarea de visualización, por lo que es necesario dar respuesta a las siguientes preguntas:

- ¿Cómo se puede identificar de forma rápida y precisa un elemento individual “objetivo” mediante el color?
- ¿Qué factores determinan si el color de un elemento “objetivo” a buscar lo hace más fácil de encontrar en relación a los colores de otros elementos “no-objetivos”?
- ¿Cuántos colores pueden ser representados a la vez para poder seguir identificando elementos “objetivo” de forma rápida y precisa?

Healey identifica tres factores separados que pueden ser utilizados para medir la efectividad de un conjunto de colores para representar datos durante la visualización científica:

- Distancia entre colores: diferencias entre colores, medidas en un modelo de color perceptivamente equilibrado.
- Separación lineal: capacidad de separar linealmente objetivos de los no-objetivos en el modelo de color utilizado.
- Categorías de colores: regiones nombradas ocupadas por objetivos como por no-objetivos.

Dados los algoritmos presentados en esta tesis, nuestro interés se centra principalmente en obtener control sobre las diferencias entre los colores. La solución más común es la de utilizar un modelo o espacio perceptivamente equilibrado o uniforme como el CIELUV, CIELab, Munsell u otros. En estos espacios, idealmente, la diferencia relativa entre pares de valores deben ser mostrados con la misma diferencia relativa de percepción del color [Hea96, HE96]. En otras palabras, estos espacios están definidos de tal manera que las diferencias perceptivas entre los colores correspondan a su distancia Euclídea en el espacio de la mejor manera posible, al menos cuando las diferencias entre los colores son pequeñas [KVK00]. Desafortunadamente, fijar la distancia entre los colores a un valor constante no garantiza que cada color vaya a resultar igual de fácil de detectar, debido a diversos factores, como por ejemplo, el contraste simultáneo [Hea96].

Lo que habitualmente llamamos color en realidad es nuestra percepción subjetiva de ondas de luz de una delgada banda de frecuencias dentro del espectro



electromagnético. Dicha subjetividad supone que las imágenes y métodos de visualización basados en el color sean dependientes del analista. Un problema similar relacionado con el uso del color en visualización por computador radica en la dependencia del dispositivo de salida. El modelo de color RGB es el utilizado por la mayoría de monitores de tubo de rayos catódicos en los que se mezclan diferentes cantidades de rojo, verde y azul para producir diferentes colores, los cuales dependen de las características físicas del monitor. Este problema se extiende también a otros dispositivos de salida como impresoras y a otros modelos de colores.

### 2.4.2.4. Marco de alineación

Es comúnmente aceptado que no todas las técnicas y formatos gráficos pueden ser aplicados apropiadamente en todas las situaciones. El término marco de alineación, aparece en la tesis de Yu [Yu95], y está basado en la idea de que una visualización de datos efectiva debe ser resultado de una combinación apropiada de técnica gráfica, objetivo de investigación y tipo de datos. A pesar de que Yu hace referencia a técnicas de AED clásicas, un análisis de estos tres aspectos proporciona un buen punto de partida a la hora de escoger y aplicar una adecuada técnica de AED. A continuación se describen estos aspectos [Yu95]:

- Taxonomía de técnicas gráficas de visualización. La complejidad del gráfico viene determinada por las siguientes características:
  - Dimensión de los datos. Es el número de variables de los datos. Los gráficos serán unidimensionales, bidimensionales, tridimensionales o multidimensionales, requiriendo mayor complejidad a medida que aumenta la dimensión. Ésta puede ser disminuida mediante técnicas de reducción de la dimensión (proyecciones lineales o no-lineales) o mediante técnicas de integración de la dimensión, que no suprimen dimensiones y visualizan todas las variables mediante un sólo objeto (caras de Chernoff, iconos basados en estrellas, etc.), véase la Sección 2.5.2.2.
  - Resolución. Se refiere al detalle de la visualización y hace referencia a la proporción de elementos originales frente a la de resúmenes o representantes de éstos. Cuando se necesita precisión, cuando el objetivo es explorar los datos o cuando se desea detectar datos atípicos, los

analistas deben visualizar todos ellos. No obstante, en muchas situaciones esto no es deseable por efectos como el ruido o por la propia técnica, por ejemplo, en un simple histograma. Por otro lado, si se desea examinar relaciones, se deben utilizar ambos tipos de técnicas de visualización.

- **Objetivos de la investigación.** Yu señala las siguientes categorías de objetivos:
  - Detectar datos atípicos.
  - Discriminar *clusters*.
  - Probar diferentes supuestos.
  - Examinar relaciones entre variables.
  - Comparar diferencias entre clases.
  - Observar procesos temporales.
- **Tipos de datos.** Existen por lo menos cuatro aspectos relevantes del tipo de datos:
  - Origen. Datos de diferentes orígenes pueden contener diferentes estructuras inherentes.
  - Formato. Nominal, ordinal, intervalos, racionales, etc.
  - Complejidad. Determinada por la dimensión, el número de observaciones y la estructura de los datos.
  - Distribución. Normal, no-normal.

### 2.5. Métodos de AED

Los métodos de AED incluyen tanto técnicas simples estadísticas como técnicas exploratorias multivariantes más sofisticadas diseñadas para identificar patrones en conjuntos de datos [Sta03a]. Las siguientes preguntas juegan un papel central a la hora de aplicar un método a un conjunto voluminoso de datos de dimensión elevada [Kas97]:

- ¿Qué tipo de estructuras puede desvelar el método?

- ¿Cómo muestra las estructuras?
- ¿Reduce la dimensión de los datos?
- ¿Reduce la cantidad de datos?

Este apartado presenta un repaso breve a las técnicas estadísticas clásicas para posteriormente hacer énfasis en varias técnicas populares de análisis exploratorio de datos multidimensionales, especialmente en métodos que ilustren estructuras en conjuntos de datos y relaciones multivariantes entre elementos o variables.

En general, los métodos tratan los datos de entrada como vectores métricos y pueden ser utilizados sin hacer ningún supuesto sobre su distribución. El estudio está más orientado hacia métodos no-supervisados (en los que se asume que no existe información *a priori* sobre la pertenencia de los datos a un conjunto de clases, ni sobre el número total de éstas). Por otro lado, cabe destacar que en aplicaciones prácticas la selección y preprocesamiento de los datos puede ser incluso más importante que el método de análisis empleado [Kas97]. En efecto, la selección o extracción de un conjunto apropiado de características discriminantes es crucial a la hora de analizar la distribución o emplear algoritmos de clasificación y reconocimiento. En cuanto al preprocesamiento, se deben considerar cuestiones como la eliminación del ruido o la normalización de los datos. La normalización llevada a cabo en esta tesis consiste en eliminar la media e igualar la varianza de cada variable (igual a uno), para que todas ellas afecten de igual modo al resultado.

### 2.5.1. Introducción a las técnicas gráficas estadísticas tradicionales

Las técnicas gráficas (estadísticas) empleadas en AED a menudo son bastante simples y pueden ser clasificadas según:

- La dimensión de los datos que tratan (unidimensionales, bidimensionales, tridimensionales o multidimensionales).
- La capacidad del usuario para manipularlas (estáticas o dinámicas).
- El tipo de tratamiento de los datos [FH02]:

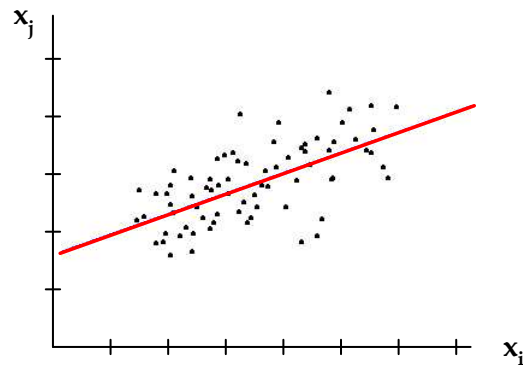


Figura 2.5: Diagrama de dispersión de dos variables, además de la correspondiente recta de regresión.

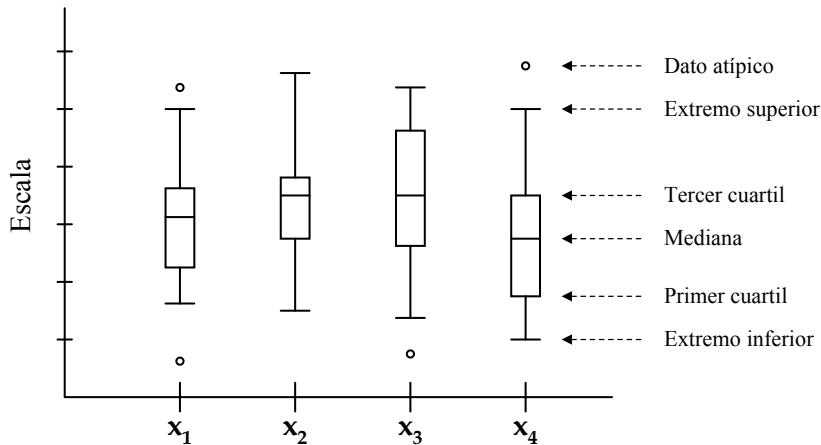


Figura 2.6: Diagramas de cajas para 4 variables.

- Representación de los datos tal y como son. Por ejemplo, los diagramas de dispersión (véase la Figura 2.5), en los que se representa dos variables de los datos, es decir, una simple proyección lineal al plano definido por dichos componentes. Otras técnicas incluyen histogramas, trazas de datos, gráficos de probabilidad, gráficos de puntos, gráficos rotacionales tridimensionales, etc.
- Representación de simples estadísticas. Por ejemplo, representación de medias, desviaciones típicas, intervalos de confianza, o diagramas de cajas (véase la Figura 2.6), en los que se representa la mediana, el primer y tercer cuartil, los valores extremos o colas de las distribuciones, y a veces otros elementos como los datos atípicos.

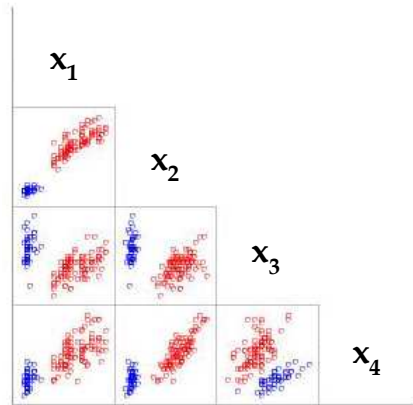


Figura 2.7: Matriz de dispersión de cuatro variables.

- Gráficos que optimizan nuestras habilidades de reconocimiento de patrones. En esta categoría caben las técnicas de análisis multivariante, en las que se hace énfasis en este apartado. Algunos ejemplos son matrices de dispersión (véase la Figura 2.7), gráficos condicionados, gráficos basados en iconos, regresión múltiple, etc.

## 2.5.2. Técnicas gráficas estadísticas multidimensionales tradicionales

### 2.5.2.1. Matrices de dispersión y *brushing*

Una de las técnicas de análisis multivariante más extendida consiste en la inspección de matrices de dispersión, en las que se muestran diagramas de dispersión de las variables de los datos tomadas de par en par, organizados de forma análoga a una matriz de correlación o covarianza. Este tipo de gráfico permite encadenar los diferentes diagramas de dispersión, de tal manera que los efectos de alterar una observación o diagrama son apreciados también en el resto (inserción, eliminación, coloración de datos, etc.).

Quizás la primera y más popular técnica explícitamente identificada como “gráfica” de AED es el *brushing*. Es un método interactivo que permite seleccionar subconjuntos de datos e identificar sus características, examinar sus efectos en relaciones entre variables relevantes (por ejemplo, en matrices de dispersión), o identificar datos atípicos [Sta03c]. La Figura 2.7 muestra una matriz de dispersión de cuatro variables en la que es posible apreciar dos grupos de datos claramente

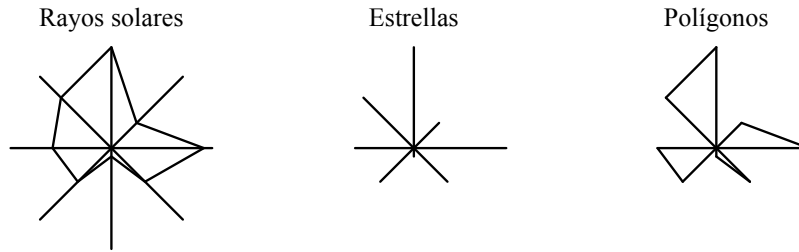


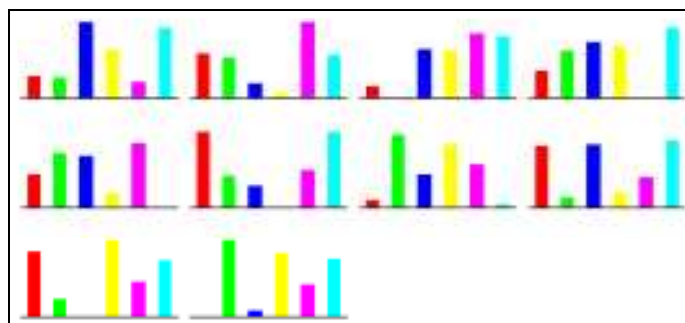
Figura 2.8: Iconos circulares.

diferenciados gracias a la aplicación de colores.

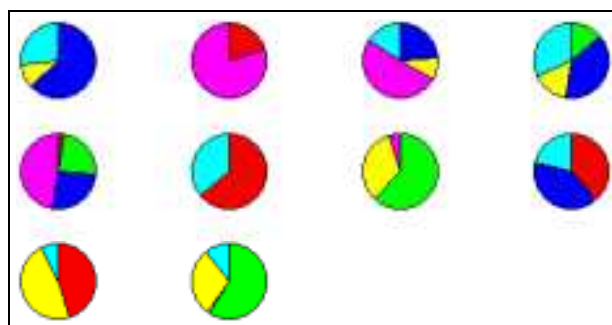
#### 2.5.2.2. Gráficos basados en iconos

Los gráficos basados en iconos pertenecen al grupo de técnicas de integración de la dimensión (véase la Sección 2.4.2.4), y permiten visualizar elementos de dimensión elevada directamente, permitiendo que cada dimensión o variable gobierne un determinado aspecto de la visualización, para posteriormente integrar los resultados en una figura [Kas97]. En otras palabras, se trata de unidades de observación representadas como símbolos multidimensionales. La idea general detrás de estos métodos es aprovechar la capacidad del ser humano para detectar “automáticamente” relaciones complejas entre múltiples variables si éstas son consistentes en un conjunto de instancias o iconos. Ofrecen una potente técnica exploratoria aunque su uso no es fácil [Sta03c]. Su mayor desventaja, en el marco de la minería de datos, es que no reducen la cantidad de datos. Si el conjunto de datos es voluminoso la representación de todos los elementos separadamente es incomprensible. Sin embargo, estos métodos pueden ser útiles para ilustrar resúmenes como vectores de referencia o centroides de clases [Kas97]. A continuación se describe una taxonomía de gráficos basados en iconos [Sta03c]:

- Iconos circulares. Siguen un formato de rueda en el que una serie de radios, asociados a cada una de las variables de los datos, emanan del centro de ésta de tal manera que los valores de las variables son representados por medio de distancias en dichos radios. Destacan los iconos basados en estrellas, iconos basados rayos solares e iconos basados en polígonos, véase la Figura 2.8.
- Iconos secuenciales. Siguen un formato más simple donde las variables son representadas mediante una secuencia de símbolos (véase la Figura 2.9, en



(a)



(b)

Figura 2.9: Iconos secuenciales de columnas (a) y sectores (b). Tomados de [Sta03c].

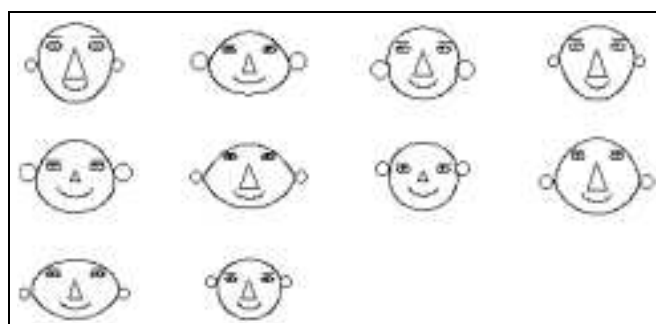


Figura 2.10: Caras de Chernoff. Tomados de [Sta03c].

la que se representan 10 elementos de seis variables). Ejemplos incluyen los diagramas de barras; iconos de columnas, líneas y perfiles; diagramas o iconos de sectores; o las curvas de Andrews [And72], en las que se utilizan los valores de las variables como coeficientes de señales sinusoidales que posteriormente son sumadas.

- Caras de Chernoff [Che73]. Es uno de los métodos gráficos más populares. Cada dimensión determina el tamaño, localización o forma de algún componente de una caricatura facial. Por ejemplo, una variable puede estar asociada a la anchura de la boca, otra a la separación de los ojos, etc. La Figura 2.10 muestra las caras de Chernoff para los mismos datos representados en la Figura 2.9. Aunque el método puede revelar patrones ocultos de interrelaciones entre las variables, resulta difícil de utilizar y requiere mucha experimentación a la hora de asignar cada variable a un determinado rasgo facial [Sta03c].

### 2.5.3. Métodos de *clustering* o cuantización vectorial

El objetivo de los métodos de cuantización vectorial (CV) o *clustering* es reducir la cantidad de datos mediante la categorización o agrupación de datos similares. Esta idea está relacionada con el modo en el que los humanos tratamos información, y una de las motivaciones para el uso de estos algoritmos es proporcionar herramientas que ayuden a construir estas categorías y taxonomías. Estos métodos pueden ser utilizados también para minimizar los efectos humanos en el proceso de agrupación [Kas97].

La aplicación más extendida de los algoritmos de *clustering* es la clasificación no-supervisada, en la que un conjunto de datos es segmentado en una serie de clases,  $C_k$ ,  $k = 1, \dots, K$ , desconocidas *a priori*. Realizan una partición del espacio de los datos en regiones asociadas a dichas clases. La manera más frecuente de realizar esta partición es mediante la definición de un conjunto de vectores representantes de cada clase  $m_k$ ,  $k = 1, \dots, K$  (también denominados vectores de referencia, vectores de medias, centroides, neuronas, etc.) y una distancia  $d$  (Euclídea, Hamming,  $L_1$ , etc.). Un dato  $x_i$ ,  $i = 1, \dots, N$ , de la distribución a analizar pertenecerá a una clase en particular si su distancia al representante de



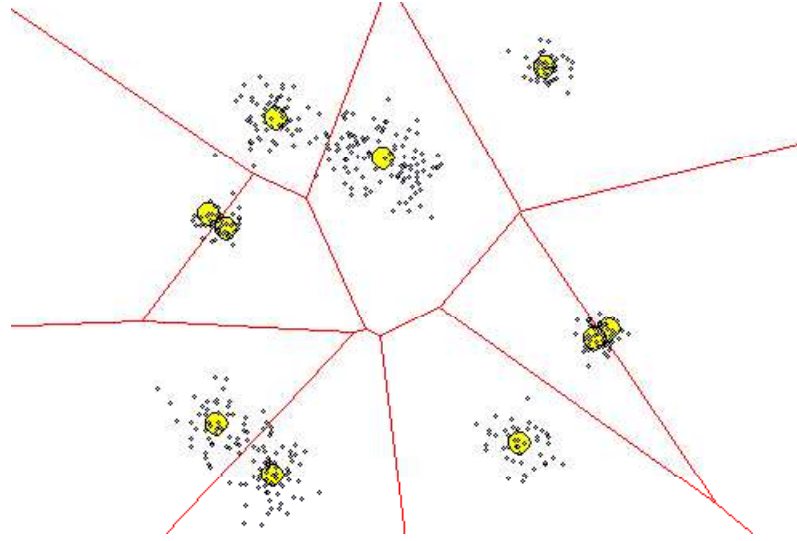


Figura 2.11: Diagrama de Voronoi, generado a partir de 10 vectores de referencia, creando una partición del espacio en clases disjuntas. Generada con [LF03].

ésta es menor que su distancia al resto de representantes:

$$x_i \in C_j \iff d(x_i, m_j) < d(x_i, m_k) \quad \forall k \neq j \quad (2.1)$$

De esta manera, la clasificación se consideraría “dura”, aunque también existen algoritmos de clasificación difusa, como el algoritmo “C-means” [Bez74], que es una generalización en la que cada muestra o dato tiene un grado de pertenencia a cada clase. El *clustering* puede estar basado también en modelos de mixturas [MB88, AG00], donde se asume que los datos son generados a partir de una serie de distribuciones paramétricas, generalmente funciones Gaussianas.

La Figura 2.11 muestra una partición (diagrama de Voronoi, en rojo) de un espacio de datos en 10 clases disjuntas, definidas mediante 10 vectores representantes (en amarillo). El diagrama de Voronoi de una colección de objetos geométricos es una partición de un espacio en celdas, las cuales consisten en los puntos más cercanos a un determinado objeto que a cualquier otro. Éstos son estudiados y aplicados en diversos campos, especialmente en geometría computacional [BKOS97].

Una definición ampliamente aceptada de una CV óptima es una partición que minimiza las distancias entre los elementos dentro de cada clase, mientras maximiza las distancias entre las clases. Sin embargo, esto da lugar a múltiples soluciones dependiendo de las distancia empleadas, véase la Tabla 2.1. La elección

Distancias dentro de una clase	$W(C_k)$
distancia promedio	$W_p = \frac{\sum_{i,i'} \ x_i - x_{i'}\ }{N_k(N_k-1)}$
distancia al vecino más próximo	$W_{vp} = \frac{\sum_i \min_{i'} \{\ x_i - x_{i'}\ \}}{N_k}$
distancia al centroide	$W_c = \frac{\sum_i \ x_i - m_k\ }{N_k}$
Distancias entre clases	$d(C_k, C_l)$
enlace simple	$d_s = \min_{i,j} \{\ x_i - x_j\ \}$
enlace completo	$d_{co} = \max_{i,j} \{\ x_i - x_j\ \}$
enlace promedio	$d_p = \frac{\sum_{i,j} \ x_i - x_j\ }{N_k N_l}$
enlace entre centroides	$d_{ce} = \ m_k - m_l\ $

Tabla 2.1: Distancias dentro de una clase  $W(C_k)$  y distancias entre clases  $d(C_k, C_l)$ ;  $x_i, x_{i'} \in C_k$ ,  $i \neq i'$ ,  $x_j \in C_l$ ,  $k \neq l$ ,  $N_k$  es el número de muestras pertenecientes a la clase  $C_k$  y  $m_k = 1/N_k \sum_{x_i \in C_k} x_i$ .

de las distancias depende de la aplicación y la norma  $\|\cdot\|$  es otro parámetro a considerar. Esta tesis utiliza la distancia Euclídea como norma y las distancias  $W_c$  y  $d_{ce}$ . Por otro lado, los métodos de *clustering* deben abordar, en general, los siguientes aspectos [VR95]:

- Determinación del número de clases.
- Eliminación de clases no utilizadas.
- Representación completa del conjunto de datos.
- División de clases existentes.

Una vez halladas las clases, la labor del analista consiste en etiquetarlas y asignarles una interpretación. Ésta puede ser difícil ya que los vectores representantes siguen siendo elementos multidimensionales [Kas97]. Por otro lado, los algoritmos suelen preferir unas formas determinadas (por ejemplo, hiperesferas), y pueden asignar datos a clases incluso cuando no existen *clusters*. Otros problemas relacionados con el *clustering* son la elección del número de clases, el cual puede ser crítico; y una buena inicialización de los centroides, que también puede ser crucial para conseguir una partición óptima o para que no aparezcan clases vacías

[Kas97]. Existen varios algoritmos incrementales (añaden centroides progresivamente) y diversas soluciones algorítmicas para tratar estos problemas: algunas basadas en estrategias competitivas [MS91, MS94, Fri94, Fri95b, Fri96]; algoritmos como el “ISODATA” [BH65], que examina los *clusters* estadísticamente en cada iteración para fusionarlos o dividirlos; el algoritmo “LBG-U” [Fri97] (véase la Figura 2.14), que es una variante del K-medias y está basada en la definición de una medida de “utilidad” de los vectores de referencia para redefinir su localización; o técnicas de optimización para evitar que el algoritmo caiga en mínimos locales, como el *simulated annealing* [VG88, FMF<sup>+</sup>89], *deterministic annealing* [BGO97], o algoritmos genéticos y técnicas de programación evolutiva [KO95, HH95], los cuales son también aplicables a MAO.

Con respecto al AED, el objetivo no es solamente la compresión de los datos, sino también hacer inferencias sobre la estructura de *clusters* y analizar si los datos exhiben una tendencia a formar dichas agrupaciones [Kas97]. A pesar de que algunos métodos de CV no proporcionan directamente visualizaciones, son utilizados frecuentemente en tareas de AED. Facilitan el uso de otras técnicas de visualización al reducir la cantidad de datos a mostrar, lo cual simplifica las imágenes, reduce el ruido y la complejidad computacional (que puede hacer prohibitivo el uso de ciertas técnicas, como la proyección de Sammon, véase la Sección 2.5.5.1). Por otro lado, existen algoritmos basados en estrategias competitivas que definen conexiones entre sus vectores de referencia para generar topologías, con las que posteriormente es posible trabajar en un espacio de dimensión menor con el objeto de realizar AED. Por ejemplo, los MAO, las “redes de representación topológica” [MS94] (que es una combinación del *neural gas* con conexiones generadas mediante el principio de aprendizaje Hebbiano competitivo [Mar93]) o los *growing cell structures* [Fri94], véase la Sección 2.5.5.5.

A continuación se describen los dos grandes tipos básicos de algoritmos de *clustering*: jerárquicos y particionales.

### 2.5.3.1. Algoritmos jerárquicos

Los algoritmos jerárquicos de *clustering* fusionan pequeños *clusters* para formar otros de mayor tamaño o dividen los mayores. De esta manera, aparecen dos tipos de algoritmos: aglomerativos o divisores, siguiendo una filosofía *bottom-up* o *top-down*, respectivamente. Los algoritmos difieren en el criterio utilizado para decidir qué *clusters* deben ser fusionados o cuál (y cómo) debe dividirse un

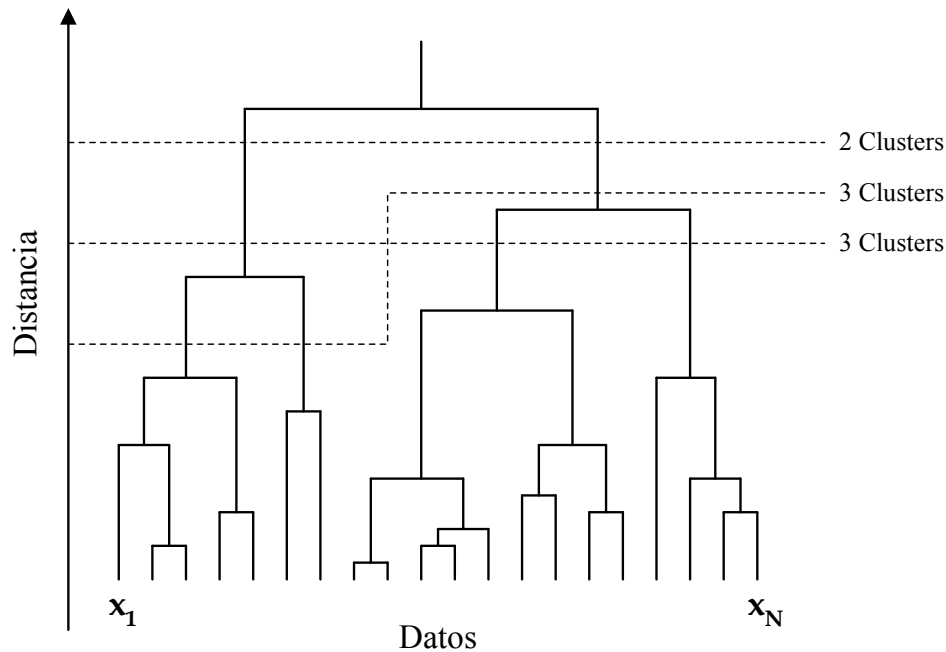


Figura 2.12: Árbol de *clustering* jerárquico o dendograma.

*cluster*. El resultado de estos procesos genera un árbol de *clustering* jerárquico o dendograma (véase la Figura 2.12), que permite visualizar las relaciones entre los diferentes *clusters*, por lo que sirve para interpretar la estructura de los datos y determinar el número de clases. El árbol puede ser “cortado” a una altura o nivel deseado, o a varios (véase [VA00]), apareciendo una partición de los datos en un conjunto de grupos disjuntos. Estos algoritmos pueden ser aplicados también a un conjunto de vectores de referencia previamente hallados, disminuyendo la cantidad de datos a representar y simplificando los dendogramas. A continuación se muestra en pseudocódigo la estructura general de los algoritmos aglomerativos de *clustering*, los cuales son más utilizados en la práctica que los divisores [VA00]:

1. Iniciación: asignar un *cluster* a cada patrón de entrada.
2. Calcular las distancias entre los *clusters*.
3. Fusionar los dos *clusters* más próximos entre sí.
4. Ir al paso 2 hasta que sólo quede un *cluster*.

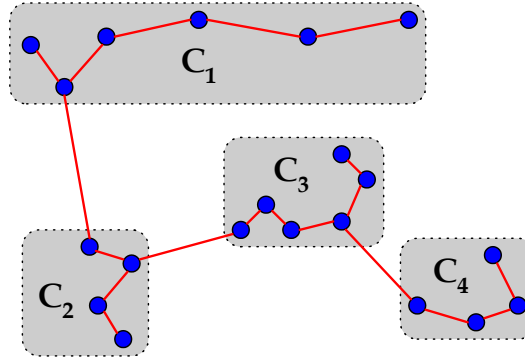


Figura 2.13: Árbol generador mínimo.

Aunque no puede ser considerado estrictamente como un algoritmo jerárquico, un caso particular de árbol de clasificación lo constituye el árbol generador mínimo (AGM) (véase [Sed83]), el cual ha sido aplicado frecuentemente en tareas de clasificación. Existen varios algoritmos para calcularlo, por ejemplo, el popular método de Prim [Pri57]. El árbol puede ser dividido según varios criterios (arista más larga, nodo con una arista significativamente más larga que el resto de las aristas incidentes en dicho nodo, etc.) dando lugar a diferentes clases. La Figura 2.13 muestra el AGM de un conjunto pequeño de datos y una clasificación de éstos en cuatro clases. Esta tesis presenta un método de visualización de MAO basado en la proyección de un AGM sobre el espacio de observación o salida de un MAO, véase la Sección 6.4.2.2.

### 2.5.3.2. Algoritmos particionales

Estos algoritmos descomponen el conjunto de datos directamente en un conjunto de  $K$  clases, generalmente minimizando algún criterio o función de error. Este criterio suele minimizar una medida de disimilitud entre las muestras de una clase, mientras maximiza la disimilitud entre las diferentes clases [Kas97]. Otra opción es hacer énfasis en la estructura local de los datos, asignando *clusters* a picos en la función de densidad de probabilidad de los datos, intentando que cada clase contenga un número similar de datos [AKCM90, SM96, Fri97, SM00]. La estructura general de un algoritmo particional es la siguiente [VA00]:

1. Determinar el número de clases.
2. Iniciar los vectores de referencia.
3. Calcular la pertenencia de los datos a cada clase.
4. Actualizar los centroides.
5. Si el algoritmo ha convergido o si la partición no ha cambiado, fin; en otro caso, ir al paso 3.

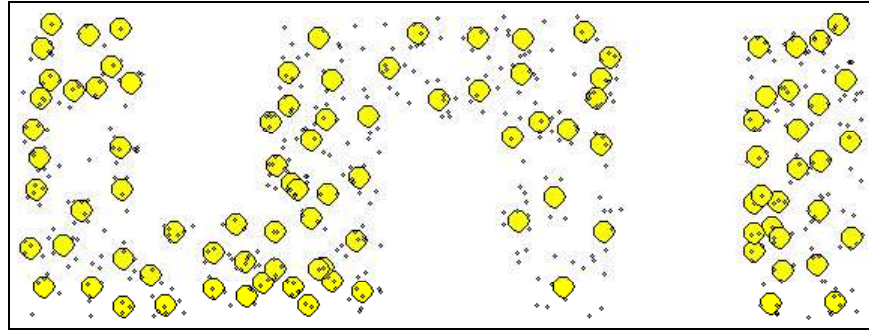
Si el número de clases es desconocido, el algoritmo puede ser aplicado repetidas veces variando el número de clases desde 2 hasta  $\sqrt{N}$ , donde  $N$  es el número total de datos [VA00]. Quizás el algoritmo particional más popular es el “K-medias” [Mac67, LBG80], el cual está estrechamente relacionado con el algoritmo de entrenamiento de los MAO. Minimiza la siguiente función de error, también denominada error de cuantización (o error de cuantización medio si es dividida por el número total de muestras, véase (3.11)):

$$E = \sum_{k=1}^K \sum_{x \in C_k} \|x - m_k\|^2 \quad (2.2)$$

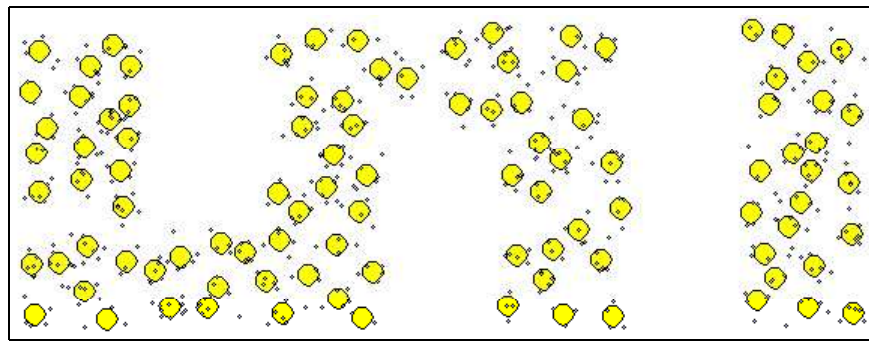
donde  $K$  es el número de clases y  $m_k$  es el centroide de la clase  $C_k$  (en los MAO  $m_k = w_k$ ).

Los métodos particionales funcionan mejor que los jerárquicos en el sentido de que no dependen de *clusters* previamente hallados. Sin embargo, los particionales hacen supuestos sobre la forma (esférica) de los *clusters* de manera implícita [VA00].

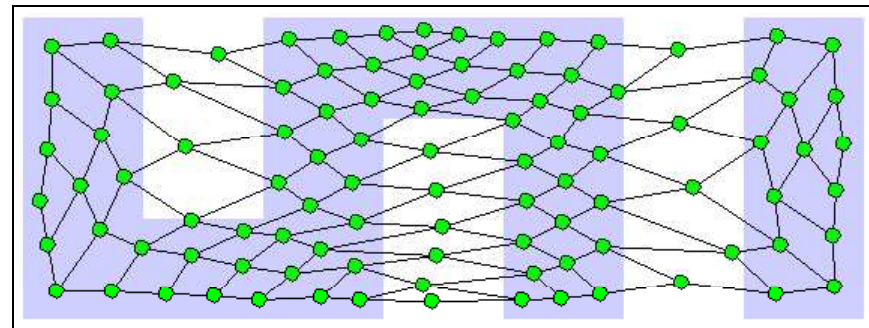
El MAO [Koh82, Koh98, Koh01], que es un tipo de RNA competitiva con entrenamiento no supervisado, es un algoritmo de *clustering* particional. También existe una serie de técnicas tradicionalmente calificadas como RNA competitivas no-supervisadas (véase la discusión sobre el uso del término “neuronal” presentada en la Sección 3.2.3.2) que son fundamentalmente algoritmos de *clustering* particional. Ejemplos incluyen redes de teoría de resonancia adaptativa [CG91], los métodos *neural gas* [MS91, MBS93] o “redes de representación topológica” [MS94], y otros algoritmos que modifican el número de *clusters* al llevar a cabo la



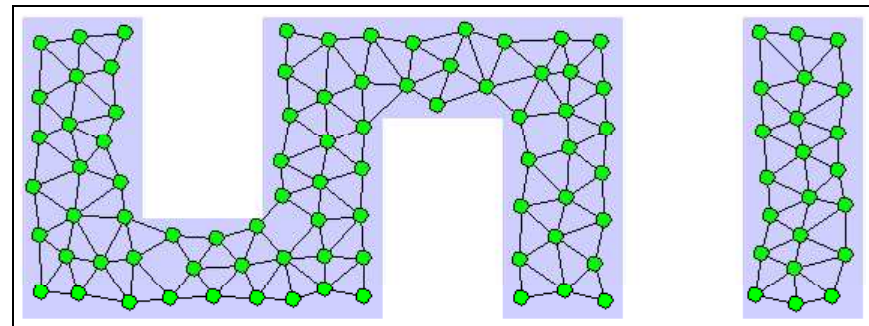
(a)



(b)



(c)



(d)

Figura 2.14: Algoritmos de CV: K-medias (a), LBG-U (b), MAO (c), *growing neural gas* (d). Generadas con [LF03].

CV, de los cuales destacan varios algoritmos propuestos por Fritzke: *growing cell structures* [Fri94], *growing grid* [Fri95a] (que es una variante incremental de los MAO) y *growing neural gas* [Fri96].

La Figura 2.14 muestra la adaptación final a una simple distribución de varios algoritmos de *clustering* utilizando 100 vectores de referencia. Es posible apreciar como el algoritmo K-medias (a) se adapta de manera menos regular a la distribución que el LBG-U (b). En este caso, el K-medias ha convergido rápidamente a un mínimo local mientras que el método LBG-U ha hallado una partición en la que cada centroide representa a un número similar de datos de la distribución. Las dos últimas imágenes de la figura ilustran un MAO (c) y el resultado de aplicar el algoritmo *growing neural gas* (d). Los dos algoritmos definen relaciones entre los vectores de referencia o neuronas vecinas generando una topología. En el caso del MAO la topología es rectangular por lo que aparecen neuronas que no se ajustan a la distribución. Sin embargo, la topología se define antes de comenzar el entrenamiento de la red y en un espacio diferente al de los propios datos, lo cual aporta ventajas a la hora de realizar AED y visualizar distribuciones de dimensión elevada. Por otro lado, en el algoritmo *growing neural gas* la topología es generada en el espacio de los datos por el principio de aprendizaje Hebbiano competitivo.

#### 2.5.4. Métodos de proyección lineales

El objetivo de los métodos basados en proyecciones es reducir la dimensión de los datos. Las proyecciones deben representar los datos o elementos de una determinada distribución de entrada de dimensión  $n$  ( $x_i \in \mathbb{R}^n$ ,  $i = 1, \dots, N$ ), en un espacio de salida ( $\mathbb{R}^m$ ) de baja dimensión ( $m \leq 3$ , por razones de visualización), de manera que se preserven los *clusters* y relaciones métricas entre los elementos lo más fielmente posible. Este apartado repasa brevemente varios métodos de proyección lineales, mientras que otros no-lineales son descritos en la Sección 2.5.5.

Una proyección lineal queda definida mediante la siguiente función:

$$y_i = A \cdot x_i, \quad i = 1, 2, \dots, N \quad (2.3)$$

los vectores fila de la matriz  $(m \times n)$ -dimensional  $A$ , también denominados vectores de dirección o proyección, definen las propiedades y características de la



proyección, y suelen ser calculados optimizando algún criterio particular para cada algoritmo de proyección. De esta manera, los algoritmos de visualización basados en proyecciones lineales constan de dos partes: la primera computa la matriz  $A$ , y la segunda, común a todos, se encarga de generar una imagen de los datos proyectados [SSS88].

Los diagramas de dispersión son las proyecciones lineales más sencillas y consisten en considerar únicamente dos componentes de los vectores de entrada. Varios diagramas de dispersión pueden ser combinados en una matriz de dispersión (véase la Sección 2.5.2.1), aunque su aplicación es limitada cuando la dimensión  $n$  de los vectores de entrada es elevada, ya que sería necesario analizar  $n(n-1)/2$  diagramas de dispersión individuales.

A continuación se describen varios algoritmos de proyección lineal utilizados para realizar AED.

### 2.5.4.1. Análisis de componentes principales

El análisis de componentes principales (ACP) [Hot33], también denominado “transformación de Karhunen-Loeve” y “proyección de autovectores”, puede ser utilizado para proyectar los datos a un subespacio del original de manera que se preserve la máxima varianza de los datos. Es un método estadístico clásico de análisis de datos y existen múltiples algoritmos para calcular la proyección, incluyendo varios basados en RNA [Oja92, CU93]. Su aplicación se fundamenta en el supuesto de que las direcciones con mayor varianza contengan la información necesaria para llevar a cabo un análisis efectivo.

El ACP realiza una rotación de los ejes del espacio original, los cuales son alineados con las direcciones de mayor varianza de los datos. La proyección considera únicamente las primeras componentes principales, que corresponden a las direcciones de mayor varianza. Sea  $\Sigma$  la matriz de covarianza de toda la distribución de datos:

$$\Sigma = E\{(x - \mu)(x - \mu)^t\} \quad (2.4)$$

donde  $E$  denota esperanza estadística,  $x$  toma valores para todos los vectores ( $N$ ) de la distribución y  $\mu$  es el centroide o vector de medias de dicha distribución ( $\mu = \frac{1}{N} \sum_{i=1}^N x_i$ ). El método la diagonaliza, obteniendo un cambio de base tras el cual los nuevos componentes aparecen incorrelados:

$$\Sigma^* = B \Sigma B^t \quad (2.5)$$

donde  $B$  es la matriz del cambio de base y  $\Sigma^*$  es la nueva matriz de correlación diagonal después del cambio de base. La matriz  $A$ , que define la proyección lineal, es formada con los autovectores (vectores fila de  $B$ ) asociados a los autovalores de mayor valor (es decir, con las direcciones de mayor varianza tras el cambio de base).

Un problema relacionado con el ACP es su dependencia de la escala de las variables o componentes, las cuales pueden ser multiplicadas por constantes sin que varíe la información contenida en los datos. De esta manera, las direcciones de mayor varianza pueden corresponder a variables definidas en una escala relativa mayor, pero cuya dirección no aporte información sobre discriminación de las clases o información relevante para el análisis exploratorio [KA96]. Para evitar este problema, en esta tesis se ha optado por normalizar los datos de manera que la varianza de todas las variables sea igual a uno.

#### 2.5.4.2. Análisis de discriminación lineal

El análisis de discriminación lineal es otra técnica estadística clásica utilizada frecuentemente en tareas de clasificación. Genera una proyección lineal de tal forma que minimiza las distancias entre los elementos dentro de cada clase, mientras que maximiza las distancias entre las clases, favoreciendo la discriminación entre las distintas clases. A diferencia del resto de métodos presentados en este apartado, es un método supervisado, por lo que requiere información sobre la pertenencia de los datos a sus respectivas clases. De esta manera, considerando un conjunto de clases  $C_k$ ,  $k = 1, \dots, K$ , con sus respectivos centroides  $m_k$ , es posible definir las siguientes medidas de dispersión:

- Dispersión dentro de una clase ( $S_w$ ). Es una medida de la dispersión de las muestras de una clase con respecto al centroide de ésta, por ejemplo:

$$S_w = \frac{1}{K} \sum_{k=1}^K E \{ (x - m_k) (x - m_k)^t \mid x \in C_k \} \quad (2.6)$$

- Dispersión entre clases ( $S_b$ ). Es una medida de la dispersión de los centroides con respecto a la media general  $\mu$  (calculada considerando todos los datos

sin tener en cuenta la división en clases), por ejemplo:

$$S_b = \frac{1}{K} \sum_{k=1}^K (m_k - \mu) (m_k - \mu)^t \quad (2.7)$$

Una vez definidas estas medidas (matrices) se busca la transformación lineal deseada  $A$  optimizando un criterio de discriminación  $J$  entre las clases basado en el cociente de dichas matrices, sus determinantes o trazas, por ejemplo,  $J = \text{tr}(S_w^{-1}S_b)$ . Cuando el cómputo de estas matrices es realizado tomando cada clase  $k$  independientemente de las demás, estas medidas son conocidas como “cocientes de Fisher” [Fis36, Día99a].

Con respecto a la visualización de datos, es posible utilizar la dirección que maximiza el cociente de Fisher, para proyecciones unidimensionales; o hallar un “plano de discriminación óptimo” [Sam70], donde se trata de hallar sucesivamente las dos direcciones que maximizan el cociente de Fisher, con la restricción de ortogonalidad para la segunda dirección una vez hallada la primera [SSS88]. Análogamente es posible generar proyecciones a un espacio tridimensional.

### 2.5.4.3. *Projection pursuit* y análisis de componentes independientes

Los métodos de *projection pursuit* son utilizados cuando no es posible obtener información *a priori* sobre la pertenencia de los datos a las clases pero se quiere encontrar una proyección discriminante (separar los *clusters* al máximo y a la vez mantenerlos compactos, es decir, la misma heurística que la sugerida por Fisher [SSS88]). Éstos buscan una proyección de los datos, generalmente a una o dos dimensiones, de tal forma que se maximice una medida “interesante”, distinta de la varianza (utilizando información de momentos de orden superior, la cual es ignorada en muchas técnicas estadísticas) e independiente de las escalas de las variables [Rip96a].

La metodología fue propuesta por Friedman y Tukey [FT74] quienes querían encontrar una medida para encontrar agrupaciones en los datos. Posteriormente, en [Hub85, Fri87, JS87] los autores se basaron en que una proyección aleatoria de los datos tiende a producir una distribución normal [DF84], por lo que argumentaron que dicha medida interesante debía ser un índice o medida de no-normalidad. Para una descripción de varios de estos índices véase [Nas92]. Los métodos se basan en su optimización, generalmente por métodos distintos a los basados en

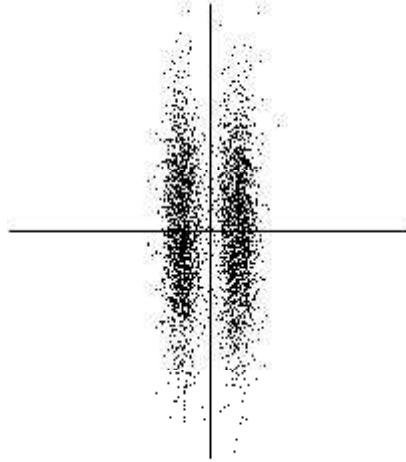


Figura 2.15: Detección de clases mediante *projection pursuit*, que proyecta los datos sobre el eje horizontal para poder discriminar las dos clases. Tomado de [Hyv99].

el gradiente, por lo que conviene que su cómputo pueda realizarse rápidamente. Otro factor importante es su independencia de valores atípicos [Rip96a]. Después de encontrar una proyección interesante, la estructura que hace que esa proyección resulte interesante puede eliminarse de los datos, y el proceso puede repetirse para buscar nuevas vistas y estructuras del conjunto de datos [Rip96a, Kas97].

La Figura 2.15 muestra una distribución formada por dos *clusters*, donde la proyección sobre el eje vertical (dirección de mayor varianza y primera que se utilizaría mediante el ACP) no conseguiría detectar la diferencia entre los *clusters*. Sin embargo, la proyección sobre el eje horizontal genera una distribución bimodal, menos cercana a una normal, en la que es posible observar claramente dos clases separadas.

El análisis de componentes independientes [Com94], es un método de transformación (proyección) lineal en el que se intenta minimizar la dependencia estadística entre los componentes resultantes. Con respecto al AED, las nuevas representaciones parecen capturar la estructura esencial de los datos en muchas aplicaciones [Hyv99]. Se trata de una técnica estrechamente relacionada con el *projection pursuit*, ya que los índices de proyección coinciden con los criterios de independencia estadística entre las variables utilizados en el análisis de componentes independientes.

Se diferencia del ACP ya que la independencia estadística es un requisito mucho más fuerte que la incorrelación. Si  $\{y_i\}$  es un conjunto de variables aleatorias

estadísticamente independientes se debe cumplir:

$$E[g_1(y_i)g_2(y_j)] = E[g_1(y_i)]E[g_2(y_j)], \quad \text{para } i \neq j \quad (2.8)$$

para cualquier par de funciones  $g_1$  y  $g_2$  (asumiendo que son medibles). Nótese que para la propiedad de incorrelación  $g_1$  y  $g_2$  son la función identidad.

A continuación se describe la relación entre independencia estadística y no-normalidad [SP98, Hyv99, HO00]. Considérese un vector de variables aleatorias  $\mathbf{y} = (y_1, y_2, \dots, y_m)$  con función de densidad de probabilidad  $f(\cdot)$ , su entropía diferencial  $H$  es definida como:

$$H(\mathbf{y}) = - \int f(\mathbf{y}) \log f(\mathbf{y}) d\mathbf{y} \quad (2.9)$$

Por otro lado, la información mutua  $I$  de las  $m$  variables aleatorias (escalares)  $y_i$ ,  $i = 1, \dots, m$ , es definida como:

$$I(y_1, y_2, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{y}) \quad (2.10)$$

La información mutua es una medida natural de dependencia entre variables aleatorias, nunca es negativa y es igual a cero si y sólo si las variables son estadísticamente independientes. Por tanto, si queremos realizar análisis de componentes independientes debemos minimizar la información mutua de las señales o componentes  $y_i$ . Por otro lado, si tenemos una transformación lineal invertible  $\mathbf{y} = \mathbf{A}\mathbf{x}$ , con  $\mathbf{y} \sim f(y)$ , y  $\mathbf{x} \sim g(x)$ , entonces:  $f(y) = g(A^{-1}y) / |\det A|$ , por tanto:

$$I(y_1, y_2, \dots, y_m) = \sum_i H(y_i) - H(\mathbf{x}) - \log |\det \mathbf{A}| \quad (2.11)$$

Los vectores  $\mathbf{x}$  corresponden a los datos originales de la distribución, y no varían. Por otro lado, imponiendo las restricciones de incorrelación y varianza igual a uno a los  $y_i$ , el determinante de  $\mathbf{A}$  debe ser constante. De esta manera, se deduce que minimizar la información mutua es equivalente a minimizar la entropía diferencial de los  $y_i$  (componentes transformados). Finalmente, como una variable normal o Gaussiana tiene la mayor entropía entre todas las variables aleatorias de igual varianza, minimizar la entropía diferencial (maximizar la independencia) de los  $y_i$  es equivalente a maximizar su no-normalidad.

Los criterios de independencia/no-normalidad utilizados en el análisis de componentes independientes (véase [Hyv99]), están basados en medidas de información mutua, divergencia de Kullbak-Leibler, estadísticas de orden superior como los cumulantes, negentropía, principio “infomax” en RNA [BS95] y criterio de máxima verosimilitud no paramétrica [PGJ92], entre otras.

Por otro lado, el análisis de componentes independientes también está relacionado con técnicas de reducción de redundancia, ACP, deconvolución ciega o análisis factorial no-normal. Su aplicación más extendida se encuentra en el AED y en la separación ciega de fuentes, en la que varias mixturas lineales (observables) de un conjunto de señales originales (no observables y supuestamente estadísticamente independientes) pueden ser tratadas para descubrir las originales. Otra aplicación del análisis de componentes independientes es la extracción de características, fundamentada en la reducción de redundancia entre las variables [Hyv99]. La Figura 2.16 muestra un ejemplo de la aplicación del análisis de componentes independientes (tomado de [KO98a]). El objetivo es encontrar factores comunes que afectan a los flujos de caja de 40 tiendas pertenecientes a la misma cadena comercial durante tres años. La primera imagen del ejemplo muestra sólo cinco flujos de caja (a). Después de preprocesar los datos se calcularon los componentes independientes. Los cinco más “interesantes” son mostrados en la segunda imagen del ejemplo (b), donde es posible observar los siguientes factores: los dos primeros siguen los cambios debidos a periodos de vacaciones, especialmente en Navidades; el último refleja una variación más lenta debido al cambio de estaciones, donde el efecto del verano es claramente visible; el tercer factor representa una variación más lenta, por ejemplo, una tendencia; el cuarto factor es diferente al resto y más difícil de interpretar (podría describir la posición de la cadena comercial respecto a la competencia).

### 2.5.5. Métodos de proyección no-lineales

Como contraste a los métodos lineales, los métodos descritos en este apartado no proporcionan una simple fórmula para transformar un dato multidimensional a su imagen en un plano o recta. De hecho, para varios métodos presentados no existe una expresión analítica de la proyección. Con los métodos lineales puede ser difícil visualizar distribuciones cuando la dimensión de los datos es elevada y/o cuando éstos contienen estructuras no lineales [Kas97]. En casos específicos la aplicación de métodos no-lineales puede ser más adecuada, aunque su uso en

## 2.5. MÉTODOS DE AED

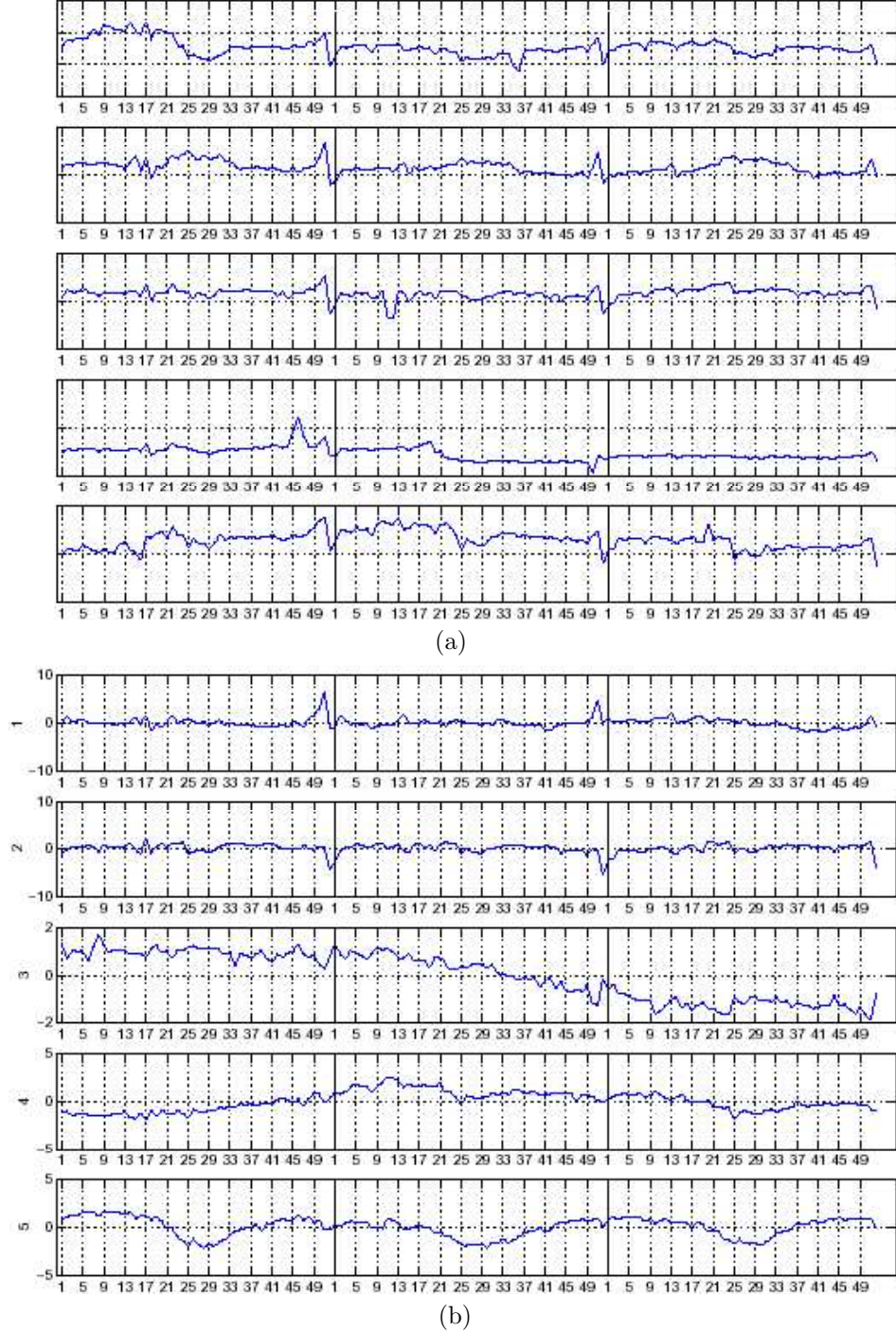


Figura 2.16: Ejemplo de análisis de componentes independientes: Flujo de caja de cinco comercios (a). Variables “independientes” que proporcionan una nueva visión e interpretación de los datos (b). Tomados de [HO00].

general es limitado [SSS88].

Varios enfoques han sido propuestos para reproducir estructuras no-lineales de dimensión elevada en espacios visibles de menor dimensión. Los más populares proyectan los datos intentando optimizar las representaciones para que las distancias entre los datos originales sean preservadas lo más fielmente posible en el plano o espacio de proyección. Los métodos difieren en los pesos que asignan a las distancias y en el modo de optimizar las representaciones [Kas97].

Entre los métodos no-lineales de proyección se encuentran las versiones no-lineales de ACP [Kra91] y análisis de componentes independientes [HP99], además de otros métodos que son descritos a continuación.

#### 2.5.5.1. Escalado multidimensional

Los métodos de escalado multidimensional (EMD) consideran medidas de disimilitud entre cada par de datos de una distribución (generalmente de dimensión elevada) que posteriormente intentan preservar en un espacio de proyección de menor dimensión. Son, por tanto, métodos de reducción de dimensión y existen dos tipos básicos: métrico, si la medida se basa en una métrica; y no-métrico, cuyas medidas no son distancias en el sentido matemático estricto.

Los métodos de EMD métrico primero consideran  $N$  puntos o datos en el espacio original de dimensión  $n$ , y seleccionan (aleatoriamente, utilizando ACP, etc.) una configuración de  $N$  puntos en el espacio de salida de dimensión  $m$  (generalmente  $m = 2$ ), que corresponde a las proyecciones de los datos originales. Posteriormente, dicha configuración es optimizada, intentando aproximar las distancias entre pares de datos en el espacio original ( $d_{i,j}^n$ ) y las distancias entre pares de datos en el espacio de salida ( $d_{i,j}^m$ ). El método original de EMD métrico [Tor52] utiliza una función de coste basada en el error cuadrático entre las distancias:

$$E_{MÉTRICO} = \sum_{i \neq j} (d_{i,j}^n - d_{i,j}^m)^2 \quad (2.12)$$

En [KVK00] se propone una variante de la ecuación anterior para conseguir un ordenamiento global de la proyección sobre un espacio de color uniforme (véase la Sección 5.5.2.2).

Quizás el método más popular de EMD métrico sea la “proyección de Sammon” [Sam69], la cual utiliza una técnica de descenso de gradiente para optimizar



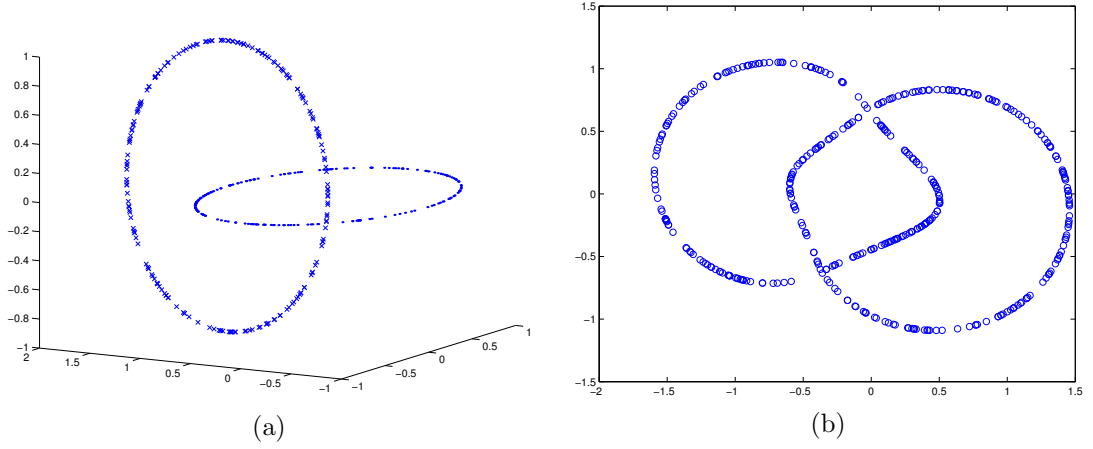


Figura 2.17: Ejemplo de la proyección de Sammon.

la configuración de las proyecciones, de tal manera que minimiza el error cuadrático medio entre las distancias. En concreto, minimiza la siguiente función de coste o “error de Sammon” (donde se ha omitido una constante de normalización irrelevante para la optimización):

$$E_{SAMMON} = \sum_{i \neq j} \frac{(d_{i,j}^n - d_{i,j}^m)^2}{d_{i,j}^n} \quad (2.13)$$

En el error de Sammon se divide por  $d_{i,j}^n$  para enfatizar la preservación de las distancias cortas. Un problema asociado con esta técnica radica en su elevado coste computacional. Una solución frecuentemente utilizada consiste en considerar únicamente el conjunto reducido de vectores de referencia provenientes de aplicar un algoritmo de CV, en vez de tratar todos los datos de una distribución. La Figura 2.17 ilustra un ejemplo de la proyección de Sammon. En (a) se muestra una distribución tridimensional compuesta por dos anillos formando una cadena, la cual es proyectada a un plano bidimensional mediante la proyección de Sammon (b).

Por otro lado, una perfecta reproducción de las distancias Euclídeas no siempre es el mejor objetivo, especialmente si las variables de los vectores aparecen en una escala ordinal. En este caso una alternativa consiste en intentar preservar el “orden de clasificación” de las distancias entre los datos. Es decir, se preserva el orden entre los datos con independencia del valor de las distancias. En este caso, el EMD es no-métrico [She62, Kru64].

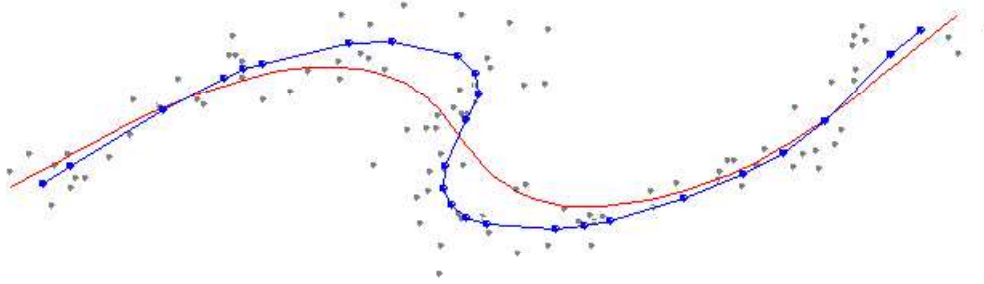


Figura 2.18: Ejemplo de curvas principales. Generada con [URLa].

Una reciente técnica de EMD no-métrico es el “Isomap” [TSL00], que está basado en las distancias entre elementos calculadas a través de un AGM. Esta técnica forma parte también de una categoría de métodos de proyección y reducción de dimensión denominada “empotrados locales” [RS00, VDG<sup>+</sup>02], los cuales intentan preservar lo mejor posible las vecindades locales de cada objeto, mientras preservan distancias globales “a través” del resto de objetos [VDG<sup>+</sup>02].

#### 2.5.5.2. Curvas principales

El ACP puede ser generalizado para formar curvas no-lineales. Mientras que el ACP construye una proyección a una “hipersuperficie” lineal, el objetivo a la hora de crear una curva principal es proyectar los datos a una hipersuperficie no-lineal. Las curvas principales [HS89] son curvas suaves definidas por la propiedad de auto-consistencia, por la que cada punto de la curva es la media de todos los puntos proyectados sobre ella. Intuitivamente éstas pasan por el medio de una nube de puntos o distribución de probabilidad  $n$ -dimensional. Aunque las estructuras extraídas son curvas, la generalización a superficies es inmediata [Kas97]. La concepción de las curvas principales es útil para entender las propiedades de los MAO, ya que las curvas principales discretas son equivalentes a éstos [MC95a]. De hecho, el algoritmo descrito en [HS89] es similar al algoritmo de entrenamiento “por lotes” de los MAO [Koh01] (véase la Sección 3.2.4.2). La Figura 2.18 muestra la curva principal según el algoritmo descrito en [HS89] (en rojo), y según una reciente variante denominada “algoritmo de línea poligonal” [KKLZ00] (en azul), que han sido ajustadas a una distribución en forma de “S” distorsionada.

Las curvas principales están relacionadas también con el método análisis de componentes curvilíneos [DH97], que es una variante de los métodos de EMD.

Utiliza un criterio para favorecer la preservación de la topología local introduciendo un factor  $F_\lambda(\cdot)$ , que es una función monótona decreciente de su argumento, en la siguiente función de coste:

$$E_{CURVILÍNEO} = \frac{1}{2} \sum_{i \neq j} (d_{i,j}^n - d_{i,j}^m)^2 F_\lambda(d_{i,j}^m) \quad (2.14)$$

### 2.5.5.3. Método de triangulación

Un problema relacionado con los métodos de EMD es su elevado coste computacional. Una manera de reducir el problema es considerar un subconjunto de las distancias entre elementos que se desea preservar. El método de triangulación [LSB77] se basa en esta propiedad, que da como resultado una proyección diferente, pero simple y eficiente. En concreto, se preservan  $(2N - 3)$  de las  $N(N - 1)/2$  distancias entre puntos. Como contrapartida, la preservación de una fracción de las distancias puede dificultar la interpretación de conjuntos extensos de datos [Kas97]. Por otro lado, el método puede ser útil en combinación con la proyección de Sammon [BJD81].

Según el método, los datos o puntos son proyectados a un plano secuencialmente, de manera que se preservan las distancias desde el nuevo punto proyectado a dos puntos previamente proyectados. Los autores del método utilizan un AGM cuyos nodos son datos proyectados y donde los pesos de las aristas son distancias entre puntos, de tal manera que se preservan todas las distancias entre vecinos más cercanos. Existen dos enfoques principales que consisten en elegir dos elementos con los que el nuevo va a preservar exactamente las distancias:

1. Un mismo elemento de referencia es escogido siempre en el espacio original de los datos, mientras que el segundo es el vecino más cercano al dato a proyectar.
2. Los elementos escogidos son los dos más cercanos al punto a proyectar.

El primer enfoque tiende a preservar información global, mientras que el segundo intenta reproducir la información local de la distribución de datos. Por otro lado, nótese que cada vez que un dato es proyectado existen dos posibles posiciones en el plano que preservan exactamente las distancias consideradas. Por este motivo es necesario utilizar un último criterio para elegir entre las dos posiciones, el cual puede estar basado en el error cometido al intentar preservar la distancia entre el nuevo punto proyectado y su siguiente vecino más cercano a considerar.

Este método está muy relacionado con el método de semejanza de triángulos, véase la Sección 6.1, para análisis visual de MAO, introducido en esta tesis. Éste proyecta cada dato en función de un conjunto fijo de elementos o puntos previamente proyectados (los vectores de referencia que son proyectados a la rejilla regular de neuronas del MAO). En concreto, intenta preservar en la proyección las relaciones entre las distancias entre el nuevo elemento y el conjunto fijo de elementos existentes en el espacio de los datos. El criterio utilizado por el algoritmo propuesto puede servir para crear nuevos métodos de proyección no-lineal si el conjunto fijo de elementos corresponde al generado por otro método de proyección diferente al MAO (véanse las líneas futuras de la tesis en el Capítulo 7).

#### 2.5.5.4. Modelos de variables latentes

Los modelos de variables latentes representan la densidad de probabilidad de datos de varias dimensiones en términos de un conjunto reducido de variables latentes, también denominadas ocultas, intrínsecas o factores. Un ejemplo es el clásico análisis factorial (véase [Día99b]), que está basado en una transformación lineal entre el espacio latente y el original. La técnica está relacionada con el ACP y es un enfoque estadístico para analizar interrelaciones entre un elevado número de variables, para poder explicarlas en términos de sus dimensiones (factores) comunes, y poder condensar la información contenida en varias variables originales en un subconjunto más pequeño de dichas dimensiones (factores), con una pérdida mínima de información.

Entre los métodos no-lineales destaca la proyección topográfica generativa [BSW98]. Es un método latente gráfico que puede simplificar la estructura de los datos proyectándolos a un espacio de variables intrínsecas [YZ01]. La proyección topográfica generativa puede ser considerada como un método de ACP no-lineal, que también está relacionado con los MAO. Al igual que éstos, proyectan los datos a un espacio observable utilizando una rejilla regular, pero seleccionan cada punto en la rejilla de manera probabilística de manera que un dato puede ser proyectado a todo el espacio latente, permitiendo una mejor visualización [YZ01].

El método define una densidad de probabilidad (discreta y uniforme para poder compararlo con los MAO, y por razones de coste computacional) en un espacio latente (por ejemplo, bidimensional) y una transformación de dicho espacio al original, generando otra distribución que puede ser modelada, por ejemplo, mediante una mixtura de funciones Gaussianas con restricciones. La transformación

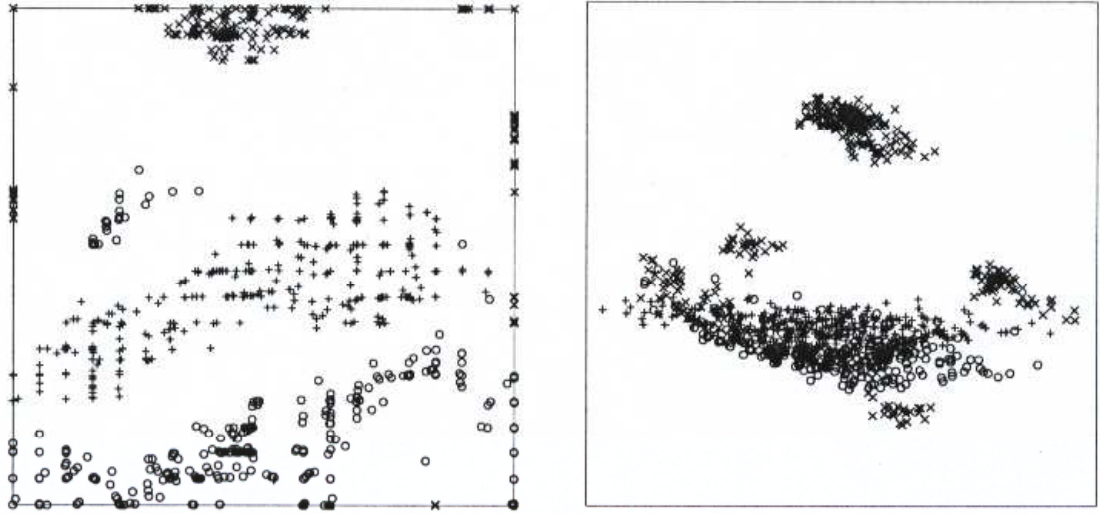


Figura 2.19: Ejemplo de proyección topográfica generativa. Tomado de [BSW98].

puede ser generada a partir de varios modelos: redes neuronales con alimentación hacia adelante, modelos de regresión lineal generalizada, etc., y el objetivo es optimizar los parámetros de la transformación (por ejemplo, los pesos y umbrales de la red neuronal) siguiendo el criterio de máxima verosimilitud. Con el objetivo de la visualización, la transformación es invertida utilizando el teorema de Bayes, dando lugar a una distribución *a posteriori* en el espacio latente.

Existe una conexión entre la proyección topográfica generativa y los MAO si consideramos los vectores latentes como neuronas y las funciones base como la fuerza de conexión entre las neuronas [KO98b]. Por otro lado, el algoritmo de la proyección topográfica generativa supera la mayoría de limitaciones de los MAO (mencionadas por el propio Kohonen [Koh01], como son: la ausencia de una función de coste, la carencia de una base teórica para escoger los parámetros y asegurar el orden topográfico, falta de demostraciones generales sobre su convergencia, y el hecho de que el modelo no define una densidad de probabilidad) sin introducir desventajas significativas [BSW98]. La Figura 2.19 muestra dos proyecciones de datos de “flujo de petróleo”: en la imagen de la izquierda se muestra la generada por la proyección topográfica generativa, la no-linealidad del método permite separar claramente los *clusters*; mientras que la imagen de la derecha corresponde a la generada mediante ACP. El ejemplo está tomado de [BSW98].

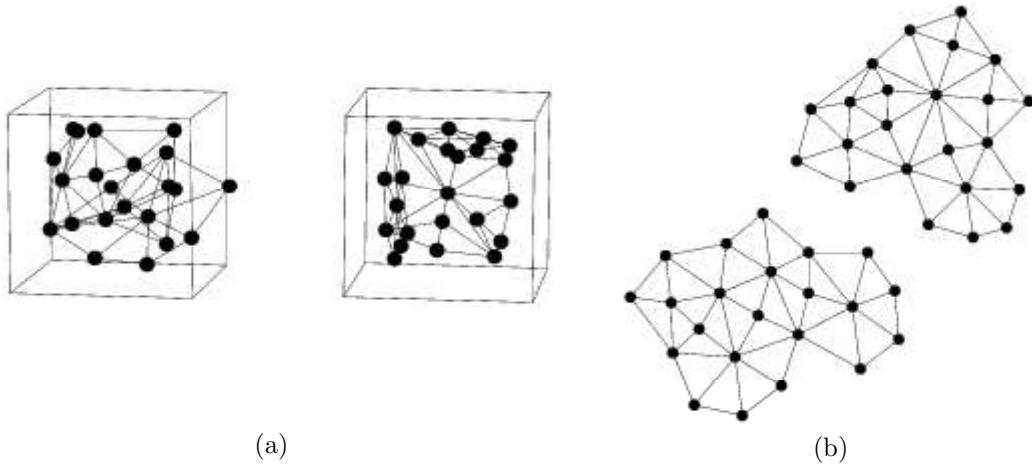


Figura 2.20: Adaptación de *growing cell structures* y su proyección a un plano. Tomado de [Fri94].

#### 2.5.5.5. Métodos basados en redes neuronales artificiales y estrategias competitivas

Existen diversas RNA diseñadas para realizar proyecciones de datos, las cuales suelen implementar métodos de proyección clásicos, véase [MJ95]. Esta última referencia bibliográfica propone varias RNA para implementar métodos clásicos como: el ACP, el análisis de discriminación lineal y la proyección de Sammon; además de proponer un método de análisis visual de MAO, que resulta ser una simple matriz de distancias (véase la Sección 5.1.2); y un último método para realizar análisis de discriminación no-lineal.

Como se comentó en la Sección 2.5.3, existen RNA competitivas que definen conexiones entre sus vectores de referencia, lo que permite generar una proyección a un espacio de dimensión menor. Las *growing cell structures* [Fri94] conectan sus células o (vectores de referencia) para generar hipertetraedros en el espacio original. El algoritmo de entrenamiento adapta dichas células a la distribución a la vez que inserta y elimina células según un criterio basado en la relativa frecuencia de señales recibidas por una célula (proporción de elementos asociados a cada célula). El algoritmo empieza con dos, o tres células formando un triángulo, y poco a poco va insertando (y eliminando) células, las cuales son situadas en los puntos medios entre dos de ellas. De esta manera, utiliza un simple modelo físico de manera que las células proyectadas se atraen y repelen en el espacio de menor dimensión hasta que el proceso queda estabilizado, mientras que las nuevas células insertadas son proyectadas también al punto medio entre dos de ellas, de

forma análoga a como se insertan en el espacio original. La Figura 2.20 muestra un ejemplo de esta proyección: la distribución empleada se compone de dos cubos tridimensionales a los que se han adaptado varias células (a); las proyecciones de éstas son mostradas en (b).

Quizás el método neuronal más popular para realizar AED sea el MAO [Koh82, Koh01]. De hecho, el método anterior puede ser visto como una variante de los MAO [Kas97, Ves00]. Los MAO realizan una CV “ordenada” de forma que proporcionan también una proyección de sus vectores de referencia o neuronas a una rejilla regular de puntos en un espacio de baja dimensión, adecuado para ser analizado por métodos de visualización. El Capítulo 5 describe los métodos más relevantes del estado del arte de análisis visual de MAO, mientras varios nuevos métodos que complementan a los anteriores y/o mejoran sus prestaciones son propuestos en el Capítulo 6.

# Capítulo 3

## MAPAS

## AUTO-ORGANIZATIVOS

### 3.1. Revisión de Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA) son técnicas analíticas modeladas según procesos de aprendizaje (hipotéticos) en el sistema cognitivo y funciones neurológicas del cerebro. En los últimos años se ha producido un aumento de interés en las RNA, las cuales han sido utilizadas con éxito en una amplia gama de áreas como economía, medicina, ingeniería, física, y otras; aplicadas a problemas de predicción, clasificación, control, optimización, aproximación de funciones, análisis de datos, entre otros (véase la Sección 3.1.5). Sus ventajas son debidas a dos factores clave [Sta03b]:

- **Potencia.** Las RNA son técnicas muy sofisticadas capaces de modelar funciones extremadamente complejas. Son métodos no-lineales, lo cual las capacita para resolver muchos problemas reales, intratables por muchas técnicas clásicas lineales.
- **Facilidad de uso.** Las RNA aprenden de ejemplos. El usuario primero recopila datos representativos para posteriormente emplear algoritmos de entrenamiento que permiten a la red aprender automáticamente la estructura de los datos. El usuario debe poseer cierto conocimiento en cuanto a la selección y preparación de los datos de entrada, el tipo de RNA a utilizar, los parámetros de entrenamiento y la forma de interpretar los resultados.



No obstante, el nivel de conocimiento necesario para aplicar RNA con éxito es mucho menor que el necesario para aplicar, por ejemplo, métodos estadísticos no-lineales tradicionales.

El conexionismo o estudio de las RNA surgió de la investigación en IA, en un intento de imitar la tolerancia a fallos y capacidad de aprendizaje de sistemas neuronales biológicos, mediante el modelado de la estructura de bajo-nivel del cerebro. De esta manera, su enfoque es diferente al de los sistemas expertos, los cuales están basados en modelos de procesos de razonamiento de alto nivel [Sta03b]. Muchos investigadores del campo de las RNA creen que éstas ofrecen la aproximación unificada más prometedora en cuanto a la construcción de sistemas informáticos verdaderamente inteligentes [Pat96]. Aunque originalmente fueron inspiradas por la neurobiología, el estudio de las RNA se ha convertido en un campo interdisciplinario, abarcando la informática, ingeniería, matemáticas, física, psicología e incluso la lingüística. El lector interesado puede encontrar más material introductorio sobre las RNA en [Fau94, Bis95, Rip96a, JMM96, Bis98, Hay99] y en los trabajos citados en este apartado.

#### 3.1.1. Inspiración biológica

El cerebro está compuesto por alrededor de 10000 millones de neuronas altamente interconectadas (desde un centenar hasta varios miles de conexiones por neurona). Una neurona es una célula nerviosa especializada que procesa información y propaga una señal electroquímica. Está compuesta de una estructura ramificada de entrada (las dendritas), otra de salida (el axón) y un cuerpo celular (soma). Una neurona recibe señales (impulsos) de otras neuronas a través de sus dendritas y transmite señales por el axón, el cual se divide en numerosas ramificaciones al final de las cuales se encuentran las sinapsis, que son las estructuras elementales y unidades funcionales entre dos neuronas. Cuando una neurona se “dispara”, envía una secuencia de impulsos por su axón que alcanza las terminales sinápticas, donde se liberan neurotransmisores en la membrana celular pre-sináptica. Éstos son difundidos por la brecha o salto sináptico y alteran el estado eléctrico de la membrana post-sináptica. En función del neurotransmisor liberado el mecanismo puede excitar o inhibir, dependiendo del tipo de sinapsis, la tendencia de disparo de la neurona receptora. Así se modifica el potencial eléctrico “de membrana” de la neurona, el cual determina su actividad. Ésta mide la

frecuencia de generación de impulsos, que es de unos 50 a unos pocos cientos de impulsos por segundo. Después de dispararse, existe un periodo refractario que dura unos 10 ms durante el cual no puede volver a dispararse. De esta manera, las señales cruzan las sinapsis y llegan hasta otras neuronas que a su vez pueden dispararse, si el total de la señal recibida supera un cierto umbral de potencial. En cuanto al aprendizaje, Hebb postula que éste consiste principalmente en alterar la “fuerza” de las conexiones sinápticas. Así, la efectividad de las sinapsis puede ser ajustada por las señales que pasan por ellas de tal manera que éstas puedan “aprender” de las actividades en las que participan. La regla de Hebb, véase [Heb49], ha servido de inspiración para muchos modelos de aprendizaje en RNA a través de los años. Desde hace más de medio siglo investigadores han propuesto modelos matemáticos para reproducir el comportamiento de las neuronas biológicas, individualmente y en grupos o redes, formando diversos tipos de RNA, véanse las Secciones 3.1.2 y 3.1.4. La Figura 3.1 presenta un esquema de una neurona y varias imágenes de neuronas biológicas obtenidas a partir de diferentes técnicas.

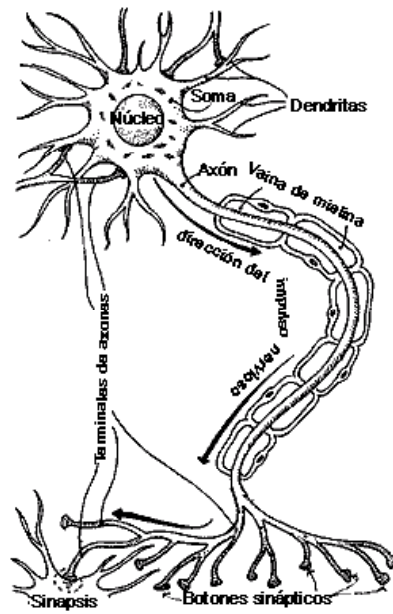
Las redes biológicas pueden procesar millones de estímulos en milisegundos. No obstante, los procesos tienen una naturaleza electroquímica, por lo que su propagación es relativamente lenta. Esto supone varios ordenes de magnitud de diferencia con respecto a las operaciones en picosegundos realizadas, en serie, por las computadoras digitales convencionales. De esta manera, dichas computadoras superan al ser humano en cuanto a computación numérica y manipulación simbólica relacionada. Sin embargo, una persona puede resolver sin dificultad ciertos problemas (por ejemplo, de percepción, como reconocer la cara de un individuo) a tal velocidad que los más modernos sistemas informáticos son ampliamente superados.

La evolución ha proporcionado al cerebro humano muchas características deseables, prácticamente ausentes en computadoras modernas. Entre ellas se encuentran [JMM96]:

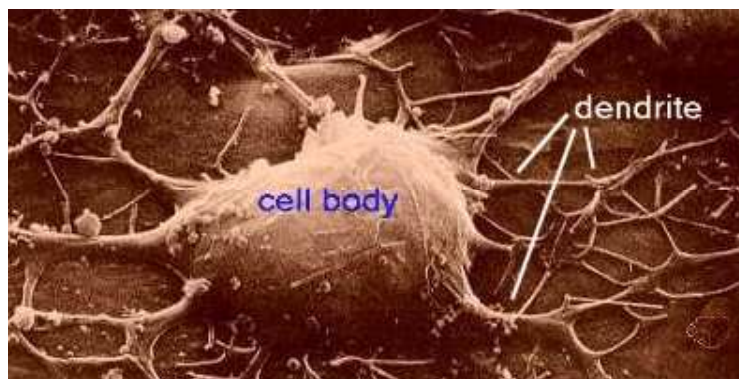
- Paralelismo masivo.
- Representación y computación distribuida.
- Capacidad para aprender.
- Capacidad para generalizar.

### 3.1. REVISIÓN DE REDES NEURONALES ARTIFICIALES

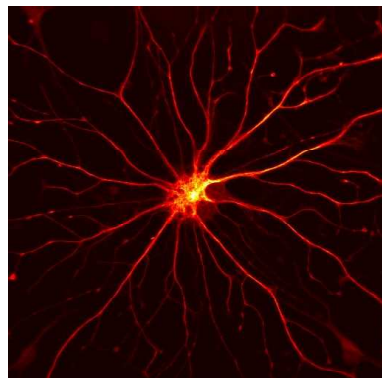
---



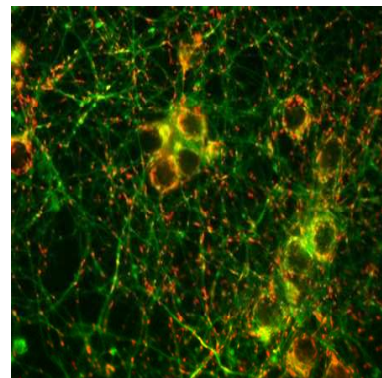
(a)



(b)



(c)



(d)

Figura 3.1: Esquema general de una neurona (a). Microfotografía electrónica de una neurona (b). Fotografías de neuronas utilizando métodos de impregnación (c) y (d). Imágenes tomadas respectivamente de [URLb, URLc, URLd, URLe].

- Capacidad de adaptación.
- Procesamiento inherente de información contextual.
- Tolerancia a fallos.
- Bajo consumo de energía.

Inspiradas en las redes neuronales biológicas, las RNA son intuitivamente llamativas al tratarse de modelos simplificados de éstas, en un intento por presentar mecanismos que exhiban algunas de las propiedades deseables citadas. De esta manera, las RNA son sistemas computacionales con paralelismo masivo de un número generalmente elevado de procesadores simples (neuronas) con muchas interconexiones, en las que se pretende utilizar ciertos principios de “organización” que se creen presentes en el cerebro humano. Por otro lado, modelar un sistema nervioso biológico utilizando RNA puede incrementar nuestro entendimiento de funciones biológicas [JMM96]. De todas formas, no hay que olvidar que muchos detalles biológicos son ignorados en los modelos artificiales (producción de secuencias de pulsos, asincronismo, los diferentes tipos, formas geométricas, periodos refractarios, el efecto de hormonas y otros productos químicos, etc.), que pueden resultar necesarios para modelar el comportamiento del cerebro. En cualquier caso, incluso los modelos simplificados poseen suficiente potencia computacional como para promover investigación muy interesante [Teb95].

#### 3.1.2. Desarrollo histórico

El estudio moderno de las redes neuronales biológicas comienza en la segunda mitad del siglo XIX, con investigadores como Camillo Golgi y Ramón y Cajal, entre otros. En esta época se produjeron avances considerables al llevarse a cabo los primeros estudios extensos del sistema nervioso humano. Por ejemplo, Ramón y Cajal acabó con la teoría reticular determinando la individualidad de las neuronas [Ram77, La202]. En las décadas siguientes, el marco de las redes fue elaborado al identificar distintos tipos de neuronas, respuestas eléctricas, patrones de conectividad y áreas funcionales en el cerebro. Aunque los neurobiólogos estudiaron con relativa facilidad la funcionalidad de neuronas individuales, fue extremadamente difícil establecer cómo éstas trabajaban en grupo para adquirir funcionalidad de alto nivel como la percepción o cognición. La introducción de las potentes computadoras hizo posible crear modelos de sistemas neuronales para experimentar

y entender mejor sus propiedades [Teb95].

La investigación relacionada con las RNA consta de tres periodos de extensa actividad [JMM96]. El primero es debido al trabajo pionero de McCulloch y Pitts [MP43], quienes propusieron el primer modelo computacional de una neurona, la “unidad binaria de umbral”, que creó gran expectación al comprobarse que podía calcular cualquier función lógica booleana básica finita. Otro de los primeros resultados fue publicado por Hebb [Heb49] al sugerir un proceso plausible de aprendizaje neuronal.

Un segundo periodo de extensa actividad investigadora fue provocado por el sensacional trabajo de Rosenblatt con los “perceptrones”, véase [Ros62]. Desarrolló variantes de éstos y estudió varias formas de aprendizaje, llegando al teorema de convergencia del perceptrón. Este descubrimiento hizo imaginar que el objetivo de la inteligencia automática estaba cerca [Teb95]. En relación con los perceptrones hay que destacar el trabajo de Widrow, quien desarrolló un elemento neuronal simple similar al perceptrón que denominó “adaline”, siendo los “madaline” redes de estos últimos. Por otro lado, Widrow y sus colegas son responsables del procedimiento supervisado de aprendizaje conocido como el *least mean square (LMS)*, también denominado regla Delta o regla de Widrow-Hoff [WH60], que posteriormente serviría de base para el algoritmo de retropropagación del error en el tiempo. Otra red de la época a destacar es la red neuronal de tiempo-retrasado [Cai61], utilizada para tratamiento de secuencias de patrones.

A pesar de los avances conseguidos, la excitación sobre las redes de perceptrones quedó seriamente mitigada cuando Minsky y Papert [MP69] propusieron argumentos sólidos argumentando que los perceptrones estaban demasiado limitados computacionalmente como para tener un uso práctico real. Esto provocó un vacío de investigación conexionista que duró casi 15 años. En los años setenta destacan los trabajos relacionados con memorias asociativas llevados a cabo por investigadores como Kohonen [Koh72], Anderson y sus colegas [ASRJ77], y el trabajo de Werbos relacionado con el perceptrón multicapa y el rápido algoritmo de retropropagación del error [Wer74].

Desde principios de la década de los ochenta, las RNA han recibido un interés renovado y considerable. Este renacimiento es debido a trabajos como el de Hopfield [Hop82], quien sugirió que una red podía ser analizada en términos de una “función de energía”. Esto condujo al desarrollo de una potente red estocástica denominada máquina de Boltzmann [AHS85]. Poco después, Parker [Par85]

y Rumelhart et al. [RMG86] popularizaron el algoritmo de retropropagación del error que podía entrenar un perceptrón multicapa para computar cualquier función deseada, mostrando que el pesimismo de Minsky y Papert no estaba bien fundamentado. Lippmann realiza una extensa recopilación de variantes del algoritmo de retropropagación del error propuesto por Rumelhart en [Lip87]. En cuanto a redes competitivas, generalmente relacionadas con métodos de CV, cabe destacar el MAO de Kohonen [Koh82, Koh98, Koh01] (que es analizado en detalle en la Sección 3.2), el método supervisado cuantización vectorial de aprendizaje [Koh90, Koh01] y las redes basadas en la teoría de resonancia adaptativa propuestas por Carpenter y Grossberg [CG91]. Otro tipo de red a destacar son las funciones de base radial [BL88], consideradas híbridos de métodos competitivos y redes con alimentación hacia adelante. Recientemente se ha propuesto la red neuronal de pulsos [MB98], que tiene en cuenta la frecuencia de llegada de las diferentes señales de entrada. Una detallada descripción histórica de los desarrollos de las RNA se encuentra en [AR88].

### 3.1.3. Fundamentos y características

Existen muchos tipos de RNA, aunque todas tienen los siguientes atributos [Teb95]:

- Un conjunto de unidades de procesamiento. Una RNA puede contener un número elevado de unidades de procesamiento simples, basadas en modelos simplificados de las neuronas biológicas. Operan simultáneamente soportando paralelismo. La computación sólo se lleva a cabo en estas unidades. En un instante dado, cada neurona calcula una función escalar de sus entradas y transmite su resultado a unidades vecinas. Suelen ser clasificadas en unidades de entrada, de salida y ocultas.
- Un conjunto de conexiones. Las unidades de una red están organizadas en una determinada topología por medio de sus conexiones o pesos, que llevan asociados un valor y determinan la reacción computacional de la red. Las diferentes topologías dan lugar a redes no-estructuradas, en capas, recurrentes o modulares.
- Un procedimiento de computación. La computación se lleva a cabo de manera síncrona o asíncrona en las neuronas aplicando una función  $f$  que depende de sus entradas  $x_i$ , sus pesos  $w_i$  y otros parámetros (por ejemplo, un

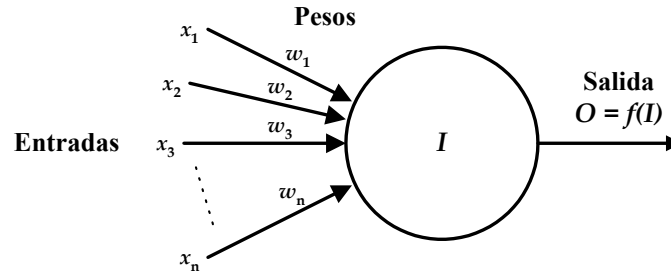


Figura 3.2: Modelo simple de una neurona artificial.

umbral). Ejemplos tradicionales son la aplicación de una función no lineal (por ejemplo, sigmoide) a una combinación lineal de las entradas y pesos, o al simple cálculo de la distancia entre el vector de entradas y el de pesos. La Figura 3.2 muestra un modelo simple de una neurona artificial.

- Un procedimiento de aprendizaje. Entrenar una red significa adaptar los valores de sus conexiones para que la red exhiba un comportamiento computacional deseado para todo patrón de entrada. Hay tipos especiales de redes que modifican también el número de unidades de procesamiento. Hay tres estrategias básicas de aprendizaje [Pat96]: supervisada, en la que un “profesor” indica a la red la respuesta correcta que debe producir; reforzada, en la que el profesor sólo indica si la red ha producido una respuesta correcta o incorrecta; y no-supervisada, en la que no interviene un profesor. A su vez, hay varios tipos de métodos de aprendizaje: Hebbiano o aprendizaje correlativo, fundamentado en la regla de Hebb; competitivo, en el que el ajuste es típicamente una forma modificada de aprendizaje Hebbiano; estocástico, por ejemplo, el procedimiento *simulated annealing* utilizado en máquinas de Boltzmann; y de corrección del error, por ejemplo, técnicas de descenso de gradiente como la regla Delta o el popular algoritmo de retropropagación del error.

#### 3.1.4. Taxonomía

Existen varios modos de clasificar las RNA. En [Pat96] se proponen tres tipos de taxonomías: según la estrategia de entrenamiento (supervisada, reforzada o no-supervisada) y el tipo de aprendizaje (corrección de error, Hebbiano, competitivo

o estocástico); según el tipo de aplicación (memoria asociativa, optimización, clasificación, reconocimiento de patrones, proyección general o predicción); y según el tipo de arquitectura (monocapa con alimentación hacia adelante, multicapa con alimentación hacia adelante, o recurrente).

En [Koh01] se sugieren las tres siguientes categorías de RNA “puras”:

- Redes de transferencia de señales. En este tipo de redes, los valores de las señales de salida dependen únicamente de señales de entrada, por lo que están diseñadas para transformaciones de señales. Ejemplos típicos son los perceptrones multicapa, madalines, funciones de base radial y redes con alimentación hacia adelante entrenadas con el algoritmo de retropropagación del error.
- Redes de transferencia de estados. En este tipo de redes la retroalimentación y las no-linealidades son tan fuertes que la actividad del estado converge rápidamente a un atractor o valor estable, al alcanzar un mínimo local de una “función de energía”. El estado final representa el resultado de la computación. Ejemplos representativos son las redes de Hopfield, máquinas de Boltzmann y memorias asociativas bidireccionales.
- Redes con aprendizaje competitivo. En este tipo de redes las células adoptan la estructuras más simples y reciben información de entrada idéntica, con la cual compiten. Por medio de interacciones laterales una célula se convierte en la “ganadora” con plena actividad y por medio de retroalimentación negativa suprime la actividad de las demás. El aprendizaje puede afectar a la neurona ganadora o a un conjunto de éstas, y al final quedan sensibilizadas a diferentes dominios del espacio de datos de entrada, actuando como decodificadores de dichos dominios. Estas redes están relacionadas con métodos de CV y destacan los MAO, redes de cuantización vectorial de aprendizaje y redes de teoría de resonancia adaptativa.

#### 3.1.5. Problemas y aplicaciones

La aplicación de RNA a un problema real requiere tener en cuenta numerosas cuestiones de diseño, como el modelo de red, su tamaño, la función de activación, los parámetros de entrenamiento, el número de muestras de entrenamiento y la preparación de los datos, donde la experiencia del usuario suele ser un factor



importante. En cualquier caso, las RNA han sido aplicadas con éxito dentro del campo de las matemáticas, estadística, medicina, economía, ingeniería, física, psicología, informática y otras ciencias. A continuación se enuncia una lista general, no exhaustiva, de los problemas tratados por las RNA:

- Predicción y pronósticos.
- Diagnósticos.
- Clasificación y categorización.
- Control y monitorización.
- Optimización.
- Análisis y minería de datos.
- Compresión de datos.
- Aproximación de funciones y proyecciones generales.
- Memorias direccionables por contenido.
- Reconocimiento de patrones.
- Percepción artificial.
- Razonamiento artificial.

Algunas aplicaciones específicas son: detección de fenómenos médicos, predicciones del mercado de valores, asignación de créditos bancarios, monitorización del estado de maquinaria, gestión de motores, reconocimiento óptico de caracteres, desarrollo de robots autónomos, transcripción de habla continua, organización de ficheros de documentos extensos, resolución del problema del viajero y muchos más. Esta tesis aborda el AED con MAO, los cuales son descritos en el siguiente apartado.

## 3.2. El Mapa Auto-Organizativo Básico

### 3.2.1. Mapas cerebrales

La corteza cerebral es posiblemente el sistema biológico más complejo. Es una capa delgada (de 2 a 3 milímetros de alto) de neuronas con aproximadamente

medio metro cuadrado de área, que está replegada adecuadamente para caber dentro del cráneo, presentando la clásica forma “arrugada”. A escala microscópica está organizada en varias capas de diferentes tipos y densidades de neuronas. A escala macroscópica consta de diversas regiones espaciales constituidas por un elevado número de neuronas similares, las cuales colaboran para realizar las tareas específicas a las que se han especializado (por ejemplo, control del habla o análisis de señales sensoriales). Se conoce desde hace tiempo que varias zonas del cerebro, especialmente de la corteza cerebral, se organizan dependiendo de diferentes modalidades sensoriales. Grupos de neuronas dentro de una región responden conjuntamente a excitaciones de las células sensoriales a las que sirven. Por ejemplo, neuronas en la corteza visual responden a ciertos patrones de luz en la retina, células de la corteza somatosensorial son excitadas por señales provenientes de células sensoriales bajo la piel y neuronas del mapa auditivo o tonotópico responden a diferentes sonidos dependiendo de su frecuencia [Pat96].

Recientes trabajos de investigación han revelado la estructura fina de varias de estas áreas, apreciándose que las señales aparecen en el mismo orden topográfico tanto en la corteza cerebral como en el órgano sensorial en el que fueron recibidas. Estas áreas o estructuras son denominadas “mapas cerebrales” [Koh01]. Por ejemplo, el mapa somatotópico es definido como una área asociativa del cerebro que realiza una proyección, que preserva la topología, de los órganos sensoriales a la corteza somatosensorial. La figura 3.3 muestra el mapa somatotópico del ser humano donde, por ejemplo, el área que corresponde a la mano está cerca de la del brazo. De hecho, una figura humana distorsionada puede ser proyectada topológicamente en dos dimensiones en la superficie de la corteza cerebral.

Los mapas de la corteza cerebral presentan las citadas propiedades topológicas según las cuales neuronas vecinas representan fenómenos o estímulos semejantes. En otros términos, se trata de una proyección “ordenada” desde un espacio de características ligado a neuronas sensoriales, a unas regiones espaciales asociadas en la corteza cerebral.

Por otro lado, una región corporal particular es representada en la corteza cerebral mediante un área proporcional al número de receptores sensoriales táctiles en dicha zona del cuerpo, y no por su tamaño. Por esta razón las neuronas forman una proyección geométrica distorsionada de la superficie del cuerpo. Nótese que las neuronas en el mapa somatosensorial no están dispuestas con forma de ser humano, pero los relativos tamaños de sus áreas indican la sensibilidad de la

### 3.2. EL MAPA AUTO-ORGANIZATIVO BÁSICO

---

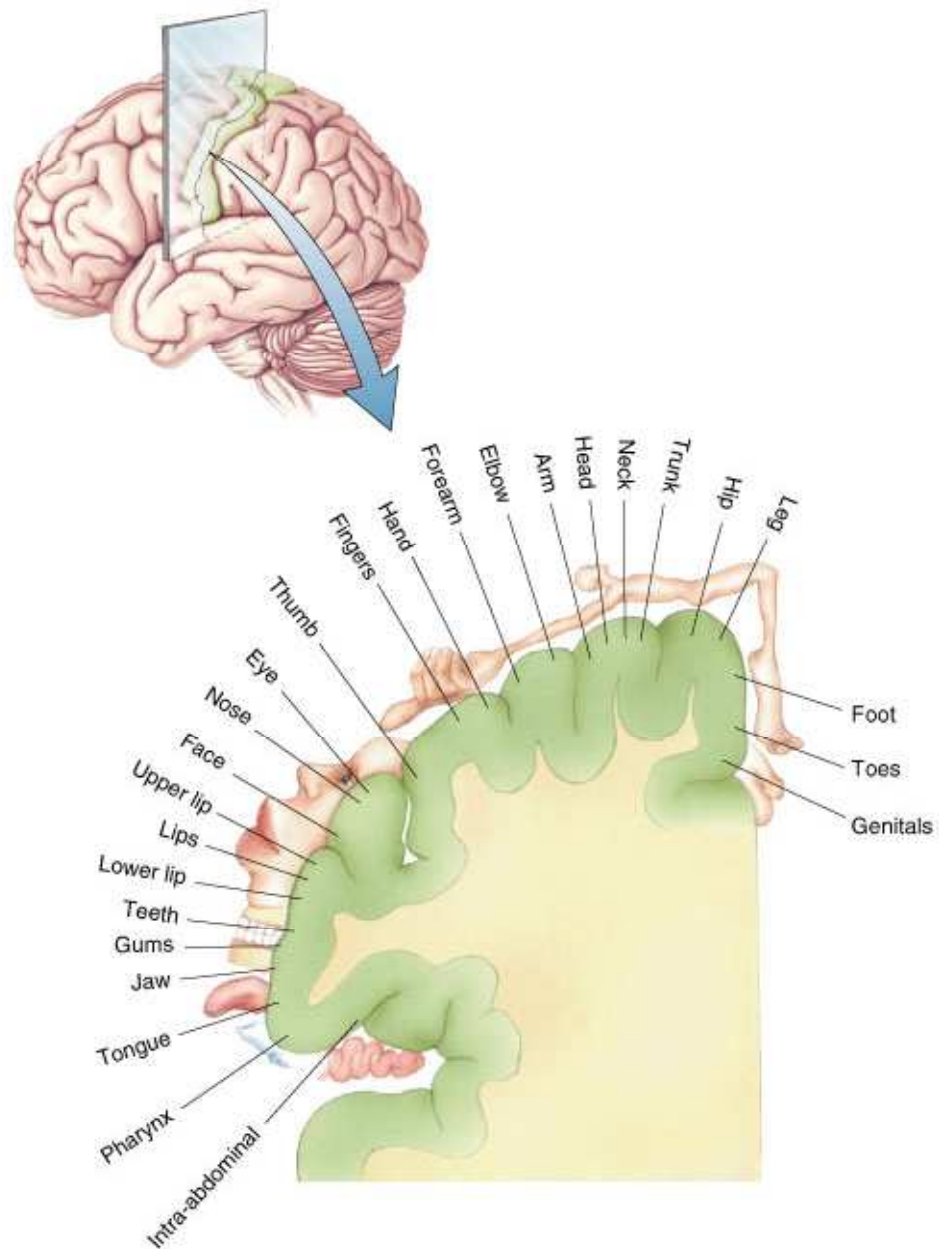


Figura 3.3: Mapa somatosensorial humano. Tomado de [URLf].

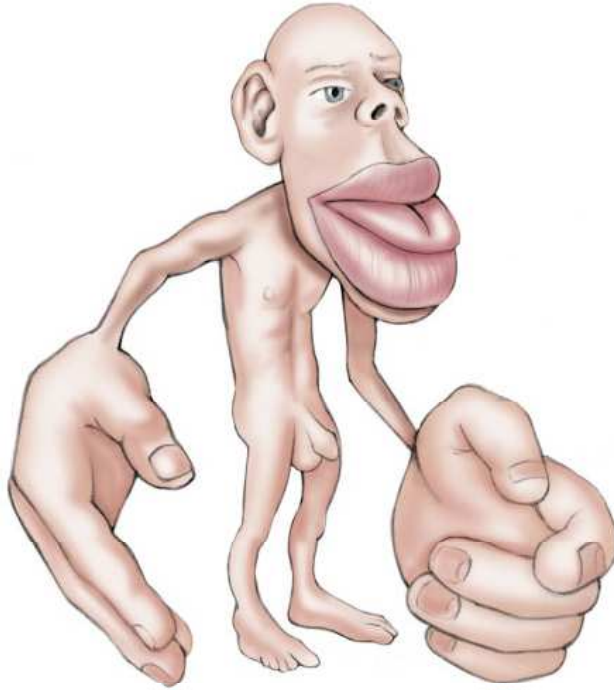


Figura 3.4: Homúnculo sensorial. Tomado de [URLf].

piel en esa zona del cuerpo. De esta manera, es posible generar un homúnculo o “pequeño hombre” sensorial, mostrado en la Figura 3.4, donde aparecen zonas muy sensibles como los labios, boca y manos. En esta tesis, esta propiedad de proporcionalidad se denomina “sensibilidad a la frecuencia”.

#### 3.2.2. Introducción y terminología

Los mapas auto-organizativos (MAO) [Koh82, Koh98, Koh01] están inspirados en la proyección y en la propiedad de organización “automática” de los mapas cerebrales, y comparten la propiedad de sensibilidad a la frecuencia. Se trata de una RNA competitiva, no-supervisada y no-paramétrica, en el sentido de que no realiza ningún supuesto sobre la distribución de datos a tratar, por lo que puede encontrar estructuras inesperadas en los datos [Kas97], encajando adecuadamente en el marco del AED.

El algoritmo de los MAO ajusta un conjunto discreto y ordenado de vectores de referencia (centroides, vectores de pesos, vectores del diccionario de bloques) a una distribución de datos de entrada vectoriales definidos en el espacio de los datos (espacio de características, espacio de entrada). Por tanto, los vectores de

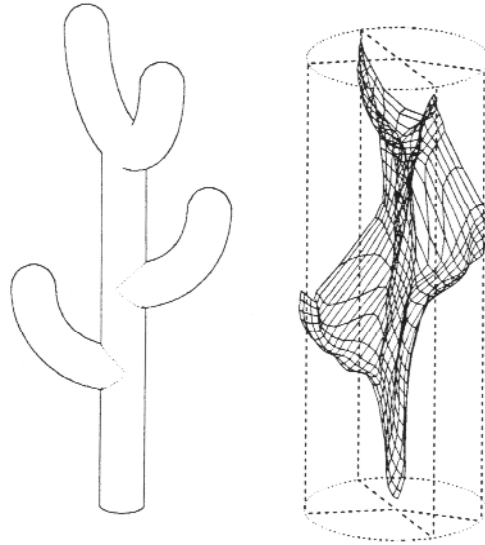


Figura 3.5: Adaptación de un MAO bidimensional a una distribución tridimensional en forma de cactus. Tomado de [Koh01].

referencia presentan la misma dimensión que los datos de entrada, y cada uno está asociado a una neurona (unidad, nodo, elemento de proceso) ubicada en una rejilla regular discreta. Dicha rejilla, habitualmente denominada “mapa”, está incluida dentro del espacio de observación (espacio de salida), que suele ser de dimensión menor o igual a tres por razones de visualización.

Los MAO pueden ser vistos como redes elásticas que se adaptan a una distribución de datos de manera ordenada, siguiendo una filosofía similar a la regresión (en lugar de ajustar una función a una distribución de datos se ajusta un conjunto de vectores de referencia), de tal forma que posteriormente puede ser posible determinar ciertas características de dicha distribución (*clusters*, distancias relativas entre clases, datos o clases atípicas, etc.) mediante técnicas de inspección visual de ciertas propiedades y medidas asociadas al mapa. La Figura 3.5 muestra la adaptación de un mapa bidimensional a una distribución tridimensional en forma de cactus, con densidad de probabilidad uniforme dentro éste. Recuperando la estructura plana inicial del mapa es posible visualizar ciertas características de los datos, por ejemplo, las diversas ramificaciones del cactus. Por tanto, resulta esencial utilizar técnicas que permitan extraer la máxima información de estas redes neuronales para una posterior visualización [RDG<sup>+</sup>02]. De hecho, el algoritmo de entrenamiento de los MAO fue desarrollado en primer lugar para la

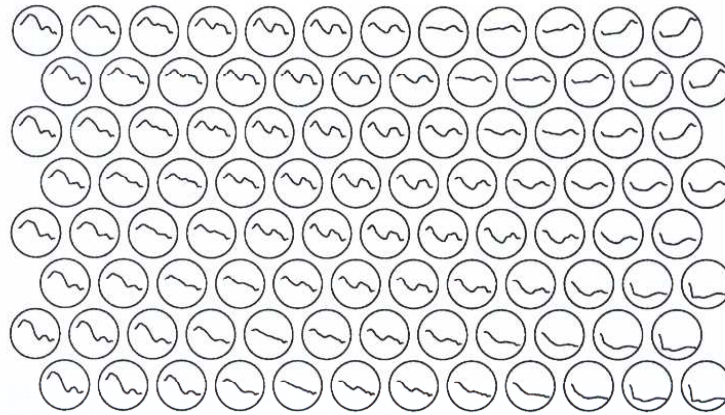


Figura 3.6: Ejemplo de un MAO con topología hexagonal que modela datos relativos al espectro de frecuencia de habla natural (finlandés). Nótese la similitud entre unidades vecinas. Tomado de [Koh01].

visualización de relaciones no-lineales de datos multidimensionales [Koh01].

Los MAO son entrenados con distribuciones de datos de entrada numéricos y pueden ser utilizados como:

- Algoritmos de *clustering*, es decir, de CV.
- Algoritmos de proyección vectorial no-lineal, mediante la correspondencia generada por el MAO entre las neuronas y los vectores de referencia asociados a ellas.

Esta dualidad es debida a la propiedad de preservación del orden topológico de la proyección (preservación de la topología), que queda reflejada en un ordenamiento de los vectores de referencia en el mapa. Aunque la lógica subyacente al modelo de los MAO es generalizable a mecanismos distintos de los neuronales [Koh82] (véase la Sección 3.2.3), es precisamente este ordenamiento el que relaciona al algoritmo de los MAO con fenómenos de organización en los mapas cerebrales, y por tanto con una idealizada estructura neuronal.

La Figura 3.6 muestra un MAO con topología hexagonal cuyas neuronas o unidades, representadas por círculos, modelan un tipo particular de espectro acústico a corto plazo de frecuencias (la curva dentro de las unidades, en la que las frecuencias bajas aparecen en su parte izquierda y las altas en la derecha), para datos de habla natural (finlandés). Inmediatamente es posible observar la similitud entre los modelos de unidades adyacentes. Por otro lado, la colección de

modelos aproxima la distribución de los espectros de entrada [Koh01].

La preservación de la topología es satisfecha cuando los vectores de referencia de neuronas próximas en el mapa, también se localizan próximos en el espacio de los datos. De esta manera, destaca la posibilidad de generar imágenes a partir de dicha proyección que revelen ciertas características inherentes sobre la estructura de la distribución de datos con la que el MAO haya sido entrenado. En la Sección 3.2.7.5 se presenta una definición más formal de preservación de la topología.

#### 3.2.3. Arquitectura

La arquitectura de los MAO es bastante sencilla. Consiste en un conjunto de neuronas  $\Xi = \{\eta_i\}$ ,  $i = 1, \dots, M$  (que pueden ser interpretadas como simples vectores  $n$ -dimensionales en vez de unidades de procesamiento), conectadas entre sí y a un vector de entrada, y organizadas geométricamente en una rejilla regular de una determinada dimensión  $m$ .

El MAO realiza una proyección no-lineal desde el espacio original de entrada ( $\mathbb{R}^n$ ) donde los datos son definidos, al espacio de salida ( $\mathbb{R}^m$ ) donde queda definida la rejilla regular. Cada neurona o nodo  $\eta_i \in \Xi$  queda representado por una posición en el mapa  $r_i \in \mathbb{R}^m$  y un vector de referencia asociado  $w_i \in \mathbb{R}^n$ . Esta asociación puede representarse mediante la definición de una correspondencia discreta uno-a-uno  $f_d$  entre el conjunto de vectores de referencia (en el espacio de los datos)  $\Omega = \{w_i\}$ ,  $i = 1, \dots, M$ , y el conjunto de las posiciones asociadas a las neuronas (en el espacio de observación o mapa)  $\Phi = \{r_i\}$ ,  $i = 1, \dots, M$ :

$$f_d : \Omega \leftrightarrow \Phi \quad (3.1)$$

Todas las neuronas están conectadas entre sí para definir vecindades y todas están conectadas a un vector de entrada  $x \in \mathbb{R}^n$ , con el cual van a comparar su vector de referencia  $w_i$  para generar un valor de salida. Lo habitual es calcular la distancia Euclídea entre ambos, aunque existen otras alternativas para la función de salida. Por ejemplo, es posible utilizar una función dependiente de valores de salida de neuronas vecinas (véase [Pat96]).

La Figura 3.7 muestra un ejemplo básico de la adaptación de un MAO bi-dimensional a una simple distribución tridimensional. Las neuronas en el mapa permanecen fijas, mientras que son sus vectores de referencia los que se ajustan, de manera ordenada, al conjunto de datos.

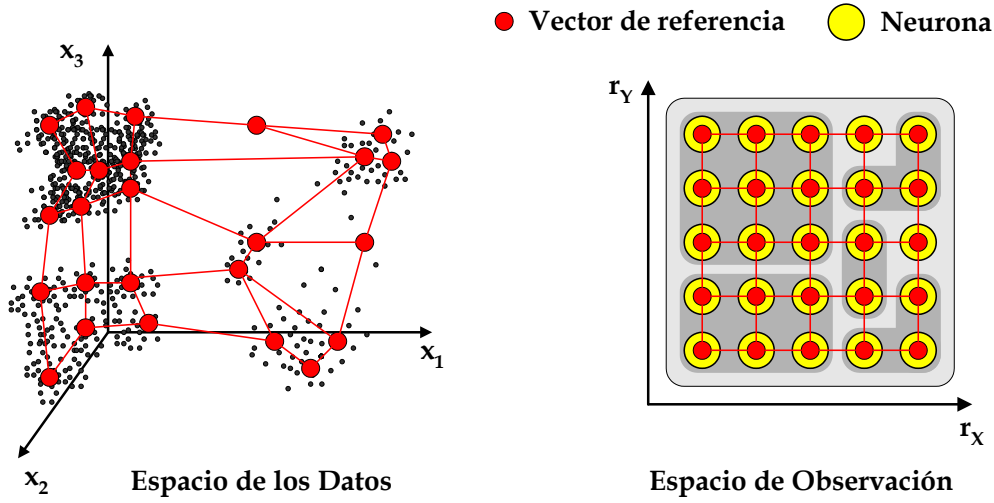


Figura 3.7: Adaptación de un MAO bidimensional a una distribución tridimensional.

### 3.2.3.1. Doble interpretación

Algunos autores describen los MAO como redes neuronales de dos capas: la primera corresponde a un dato o vector de entrada, mientras que la segunda corresponde al mapa o rejilla de neuronas. En este marco, los pesos de las conexiones entre las capas corresponden a los componentes de los vectores de referencia, mientras que la función de activación genera la correspondiente salida. De esta manera, el modelo de neurona es idéntico al representado en la Figura 3.2. En la Figura 3.8 se muestra esta primera interpretación de la arquitectura de los MAO, donde  $x \in \mathbb{R}^n$  representa un vector de entrada, y las neuronas quedan definidas mediante una posición  $r \in \mathbb{R}^2$ , un vector de referencia o de pesos  $w \in \mathbb{R}^n$  y una función de activación  $f$  común, generalmente  $f(w, x) = \|w - x\|$ , donde  $\|\cdot\|$  denota distancia Euclídea.

Por otro lado, teniendo en cuenta la filosofía de los algoritmos de CV, los MAO pueden ser vistos como un simple conjunto de vectores  $n$ -dimensionales situados en una rejilla regular  $m$ -dimensional (véase la Figura 3.9).

### 3.2.3.2. Discusión

Como se ha visto en el apartado anterior, los MAO pueden ser estudiados desde un punto de vista distinto al neuronal. Dada la simplicidad de los modelos artificiales de las redes neuronales, una interesante discusión surge al abordar el problema de definir qué es una RNA. En [Rip96b] se comenta que varios trabajos



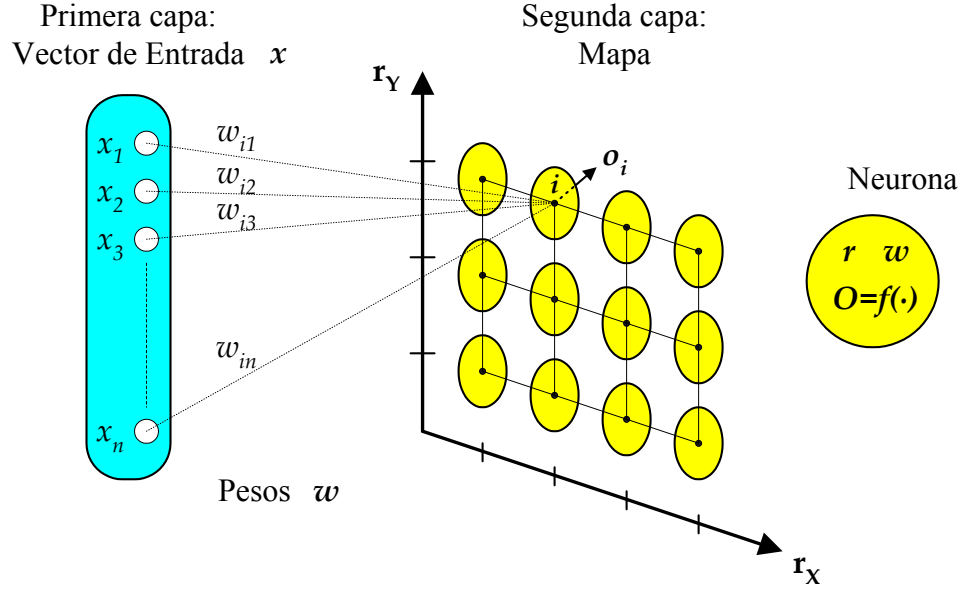


Figura 3.8: Interpretación de la arquitectura de un MAO bidimensional desde el punto de vista clásico de RNA organizadas por capas.

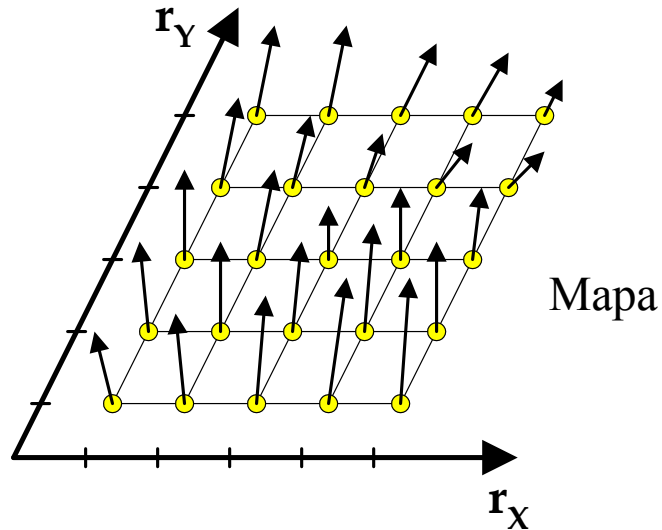


Figura 3.9: Interpretación de la arquitectura de un MAO bidimensional desde el punto de vista de los algoritmos de CV.

publicados en la literatura sobre RNA no parecen ni neuronales ni redes, en el sentido de que carecen de una motivación biológica y la información no aparece codificada únicamente en fuerzas de conexión entre neuronas. De esta manera, critica a los métodos de Kohonen (MAO y cuantización vectorial de aprendizaje, véase la Sección 3.3.4), ya que los pesos de la red no representan fuerzas de conexión. Sin embargo, Kaski [Kas97] comenta que los componentes de los vectores de referencia de los MAO corresponden a “pesos sinápticos”, generando la consiguiente contradicción. Por otro lado, dado que en la interpretación neuronal las neuronas sólo calculan distancias y no se comunican (salvo en el caso particular de generar salidas dependientes de varias neuronas), es necesario la incorporación de un proceso “superior” no-neuronal para que realice varias tareas del proceso de entrenamiento (véase la Sección 3.2.5).

Para el autor de esta tesis, los MAO pueden ser considerados “neuronales” dada la característica de organización de las neuronas artificiales en un espacio “físico” y distinto del de los datos (el mapa). Ésta resulta similar al modo en que aparecen organizados los mapas cerebrales, donde las neuronas de la corteza cerebral (física) se organizan y representan fenómenos del mundo externo (espacio de los datos).

En este sentido, y considerando las observaciones de Ripley, parece más conflictivo aplicar el término “neuronal” a varios procedimientos de CV que definen conexiones entre sus vectores de referencia para “organizarlos” únicamente en el espacio de los datos, como los varios algoritmos propuestos por Fritzke [Fri94, Fri95b, Fri96] o la combinación del método *neural gas* [MS91, MBS93] con el principio de aprendizaje Hebbiano competitivo [Mar93].

### 3.2.3.3. Topología

Esta tesis analiza principalmente mapas bidimensionales ( $m = 2$ ) de topología rectangular. Una opción frecuentemente utilizada consiste en emplear una topología hexagonal, que resulta más efectiva para el análisis visual [KHKL96]. Sin embargo, se ha optado por trabajar con rejillas rectangulares por varios motivos:

- Los estudios llevados a cabo sobre MAO no sólo se han centrado en técnicas de visualización, por lo que se ha aprovechado una infraestructura de código fuente basado en mapas rectangulares.

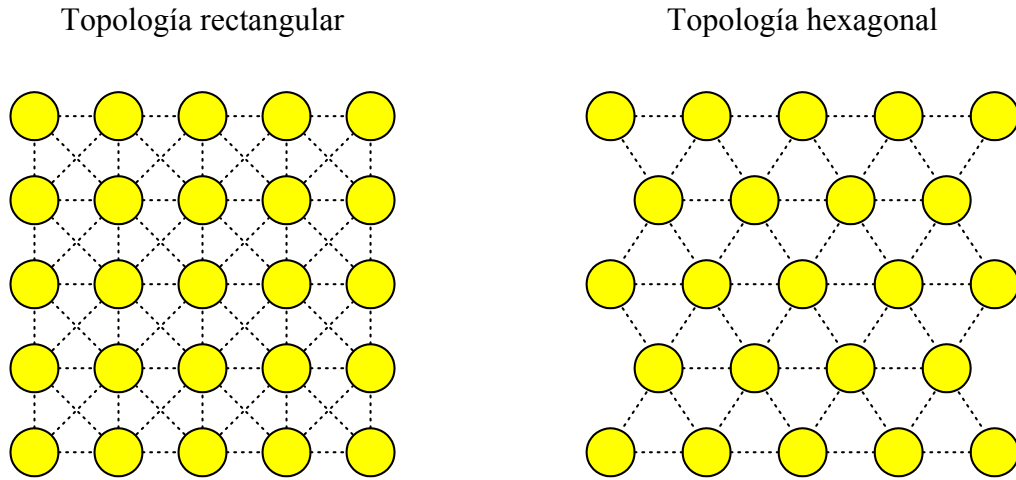


Figura 3.10: Diferentes topologías para mapas bidimensionales.

- Facilidad y eficiencia a la hora de implementar y probar los nuevos algoritmos, acelerando el proceso de investigación.
- Los algoritmos de visualización presentados son independientes del tipo de topología, y perfectamente adaptables al caso hexagonal.

En cualquier caso, en la práctica es aconsejable utilizar los nuevos algoritmos también con mapas de topología hexagonal. La Figura 3.10 muestra los dos tipos de topologías (rectangular y hexagonal) para mapas bidimensionales.

#### 3.2.4. Algoritmo de entrenamiento

El algoritmo de entrenamiento de los MAO forma parte de los métodos de “aprendizaje competitivo”. Este término hace referencia al proceso adaptativo por el cual las células de una red neuronal se especializan progresivamente para representar una serie de categorías presentes en conjuntos de datos. A medida que avanza el proceso de entrenamiento aparece una especie de división de tareas, según se van especializando las diferentes neuronas para representar diferentes tipos de entradas. Esta especialización es generada mediante una competición entre las neuronas, de manera que cuando se trata un nuevo dato, la neurona (o un conjunto de éstas) que mejor representa al dato gana y se le permite “aprender” a representarlo mejor.

Cuando existe un ordenamiento de las neuronas, como es el caso en los MAO, el aprendizaje se generaliza para que aprenda no sólo la neurona ganadora, sino

también un conjunto de sus vecinas (aunque la capacidad de aprendizaje de una neurona suele disminuir según se aleja de la neurona ganadora). El resultado final resulta en que neuronas próximas en el mapa se especializan en representar datos similares, por lo que dichas representaciones aparecen ordenadas. Esta es la esencia del algoritmo de aprendizaje de los MAO [Kas97].

El algoritmo de entrenamiento es un proceso iterativo que crea una proyección no-lineal del espacio de los datos al mapa o rejilla discreta intentando preservar la topología de los datos originales. Después del entrenamiento las neuronas pueden ser consideradas como vectores de referencia de un algoritmo de CV, y es posible aprovechar el ordenamiento entre dichos vectores para visualizar características de los datos (esta tesis propone nuevos métodos para este propósito). En cuanto a la proyección, la imagen de un dato puede corresponder al lugar de la rejilla donde se encuentra la neurona que mejor lo representa, o puede emplearse una estrategia de interpolación para situarlo en el subespacio continuo en el que se sitúan las neuronas (véanse las Secciones 5.2.3 y 5.4.4, y el nuevo método presentado en la Sección 6.1).

Un detalle importante consiste en distinguir entre “algoritmo de entrenamiento” y “proceso de entrenamiento” (véase la Sección 3.2.5). El proceso de entrenamiento ejecuta el algoritmo de entrenamiento varias veces con diferentes parámetros, además de encargarse de una serie de tareas relacionadas con varios parámetros de entrenamiento y características generales de la red, e incluso de los propios datos. Por ejemplo, se debe:

- Estudiar la dimensión intrínseca de los datos.
- Elegir una adecuada dimensión, topología y tamaño para la red.
- Especificar los parámetros de entrenamiento: tasa de aprendizaje inicial, radio de vecindad inicial, funciones decrecientes relacionadas con la tasa de aprendizaje y el radio de vecindad, número de elementos a tratar, etc.
- Asignar unos valores iniciales a los vectores de referencia de las neuronas.

Este apartado describe los dos tipos de algoritmos de entrenamiento propuestos para construir un MAO básico: secuencial, más apropiado para ser modificado y para entender la filosofía de los MAO; y “por lotes” o *batch*, similar al algoritmo K-medias, que proporciona una alternativa muy eficiente para construir MAO.

### 3.2.4.1. Entrenamiento secuencial

El algoritmo básico de entrenamiento secuencial consiste en los siguientes pasos:

1. Introducir un nuevo vector de entrada y calcular la neurona ganadora.
2. Actualizar los vectores de referencia de la neurona ganadora y sus vecinas hacia el vector de entrada, según una tasa de aprendizaje y un núcleo de vecindad centrado en la neurona ganadora.
3. Si se ha procesado un número determinado de muestras  $T$ : fin; en caso contrario, reducir la tasa de aprendizaje y el radio de vecindad, e ir al paso 1.

El número total de iteraciones del algoritmo  $T$  (número de muestras que son procesadas) es uno de los varios parámetros que deben ser elegidos antes de ejecutar el algoritmo. La Sección 3.2.5 aborda la elección de los parámetros de entrenamiento.

El primer paso consiste en presentar de un nuevo dato o vector de entrada  $x(t) \in \mathbb{R}^n$ , escogido aleatoriamente de la distribución de entrenamiento  $\Delta = \{x_i\}$ ,  $i = 1, \dots, N$ , en el instante de tiempo discreto  $t$  correspondiente a una iteración del algoritmo. Posteriormente es comparado con todos los vectores de referencia  $w_i(t)$ ,  $i = 1, \dots, M$ , del MAO para calcular la “neurona ganadora” o unidad de máxima similitud  $\eta_c(t)$ . Su vector de referencia  $w_c(t)$  es el que mejor representa al dato  $x(t)$ . Generalmente, se emplea una métrica como medida de similitud (o disimilitud) entre el dato de entrada y los vectores de referencia:

$$\|x(t) - w_c(t)\| = \min_i \{\|x(t) - w_i(t)\|\} \quad (3.2)$$

En esta tesis  $\|\cdot\|$  denota distancia Euclídea.

El segundo paso consiste en la modificación de los vectores de referencia de la neurona ganadora y un conjunto de sus vecinas para que representen mejor al dato de entrada  $x(t)$ . El procedimiento consiste en mover dichos vectores en dirección hacia  $x(t)$ . La Figura 3.11 ilustra este proceso. La fuerza de atracción

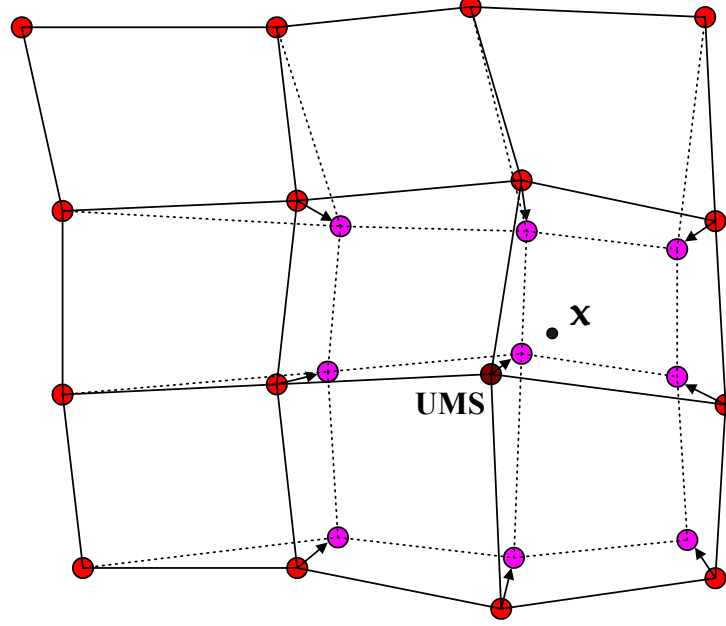


Figura 3.11: Adaptación del vector de referencia de la neurona ganadora (unidad de máxima similitud) y los de sus vecinas hacia el patrón de entrada  $x$ . Las líneas continuas y discontinuas corresponden a la situación antes y después de la adaptación, respectivamente.

de  $x(t)$  sobre el vector de referencia de una determinada neurona depende de una tasa de aprendizaje, de la distancia desde dicha neurona a la ganadora en el mapa y de la distancia desde  $x(t)$  al vector de referencia en el espacio de los datos. Durante el proceso de aprendizaje los vectores de referencia son actualizados de acuerdo con la siguiente regla:

$$w_i(t+1) = w_i(t) + \alpha(t)h_{c,i}(t)[x(t) - w_i(t)] \quad (3.3)$$

donde el subíndice  $c$  representa a la neurona ganadora para el dato de entrada  $x(t)$ . Nótese como el vector  $w_i(t)$  es adaptado hacia  $x(t)$  según la distancia que los separa y por el valor definido por el producto  $\alpha(t)h_{c,i}(t)$ , que debe ser menor o igual a uno y decrecer (tender a cero) a medida que avanza el tiempo y las muestras son procesadas.

La “tasa de aprendizaje”  $\alpha(t)$  es una función monótona decreciente que tiende a cero. Su elección no es crítica [KHKL96, Koh01] y puede ser lineal, exponencial o inversamente proporcional a  $t$ . En [MC95b] se propone la definición de una tasa de aprendizaje “óptima”, en el sentido de asegurar que todos los datos contribuyan por igual a la configuración del MAO, independientemente del orden

de presentación de éstos. En esta tesis se han empleado las siguientes tasas de aprendizaje:

$$\alpha(t) = \alpha(0) \cdot \left(1 - \frac{t}{T}\right) \quad (3.4)$$

y

$$\alpha(t) = \alpha(0) \cdot \exp(-\beta_T t) \quad (3.5)$$

donde el parámetro  $T$  representa el número de iteraciones que efectúa el algoritmo;  $\alpha(0)$  es el valor inicial de la tasa de aprendizaje, que debe ser elevado (cerca de la unidad [Koh01]) en la fase de ordenamiento si los vectores de referencia no están ordenados, y progresivamente debe descender en las siguientes fases del proceso de entrenamiento. La constante  $\beta_T$  debe ser elegida de tal forma que  $\alpha(T) \simeq 0$  en (3.5).

El denominado “núcleo de vecindad”,  $h_{c,i}(t)$ , es una función definida sobre la rejilla regular del mapa, y define el conjunto de neuronas alrededor de la ganadora que van a ser actualizadas, además de contribuir (junto con la tasa de aprendizaje) a establecer la fuerza de atracción de sus vectores de referencia hacia el dato de entrada. Por otro lado, del núcleo de vecindad define la “rigidez” de la “red elástica” del MAO en el espacio de los datos [KHKL96]. Generalmente depende del tiempo y de la distancia Euclídea entre una neurona  $\eta_i$  y la ganadora  $\eta_c$  en el mapa ( $\|r_c - r_i\|$ ), y tiende a cero según aumenta ésta. También depende implícitamente de un “radio de vecindad”  $\sigma(t)$ , que es otra función monótona decreciente y puede ser definida de forma análoga a la tasa de aprendizaje. En este caso, la elección del radio de vecindad inicial  $\sigma(0)$  es crucial en el entrenamiento (véase la Sección 3.2.5.4). Éste debe ser elevado (por lo menos la mitad del diámetro del mapa) cuando los vectores de referencia están desordenados, pequeño en la fase de adaptación y prácticamente nulo en la fase de convergencia, para que sólo aprenda la ganadora, en cuyo caso el algoritmo se comporta como el K-medias (véase la Sección 3.2.5).

Existen dos tipos clásicos de núcleos de vecindad: binario y continuo. El núcleo de vecindad binario, también denominado “burbuja” o “truncado”, define un conjunto de neuronas  $\Xi_c(t)$  alrededor de la ganadora  $\eta_c$  en función de una distancia (no necesariamente la Euclídea) y del radio de vecindad  $\sigma(t)$ :  $\Xi_c(t) = \{\eta_i \in \Xi \mid \|r_c - r_i\| \leq \sigma(t)\}$ . De esta manera, el núcleo de vecindad queda definido

mediante:

$$h_{c,i}(t) = \begin{cases} 0 & \text{si } i \notin \Xi_c(t) \\ 1 & \text{si } i \in \Xi_c(t) \end{cases} \quad (3.6)$$

El segundo tipo de núcleo de vecindad más suave y muy popular es similar a una función Gaussiana:

$$h_{c,i}(t) = \exp\left(-\frac{\|r_c - r_i\|^2}{2\sigma^2(t)}\right) \quad (3.7)$$

En este caso, la fuerza de adaptación de una neurona es mayor cuanto más cerca esté situada de la ganadora. En los trabajos presentados en esta tesis se ha empleado este tipo de núcleo de vecindad. Por otro lado, es posible variar la función anterior para que anule la actualización de los vectores de referencia cuyas neuronas estén situadas fuera del radio de vecindad.

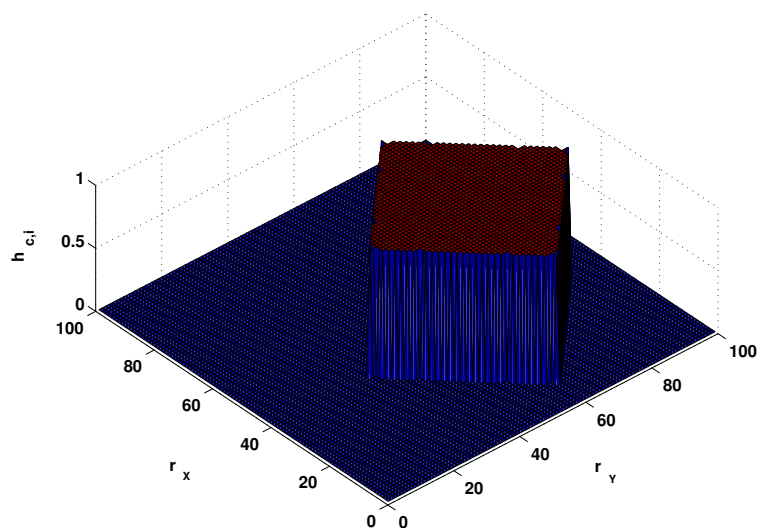
La Figura 3.12 muestra los dos tipos de núcleos de vecindad: “burbuja” (a), en el que se ha empleado la distancia  $\mathcal{L}_1$ ; y “Gaussiano” (b). En ambos casos el MAO es  $(100 \times 100)$ -dimensional, la neurona ganadora está situada en la posición (40,60), y el radio de vecindad igual a 30.

#### 3.2.4.2. Entrenamiento por lotes

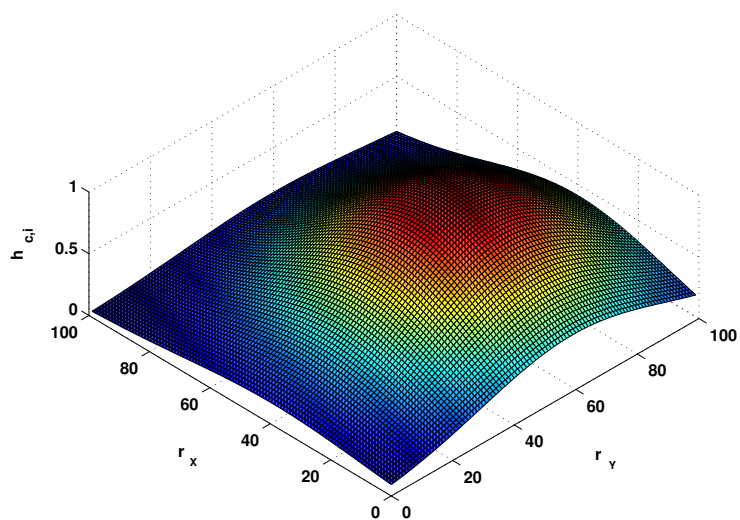
El mapa por lotes [Lut90, Koh93] es un algoritmo muy similar al K-medias, con la excepción de que introduce una vecindad decreciente con el tiempo. Utiliza todos los datos en cada iteración, por lo que es independiente del orden de presentación éstos, que afecta al proceso secuencial. Al no definir una tasa de aprendizaje no sufre problemas de convergencia y presenta valores asintóticos más estables para los  $w_i$  que los del algoritmo secuencial. Es especialmente efectivo si los vectores de referencia iniciales están algo ordenados, incluso si éstos no se han adaptado todavía a la distribución [Koh01]. En experimentos empíricos el tiempo de entrenamiento es reducido con respecto al empleado por el algoritmo secuencial [MC95a].

A continuación se describe el algoritmo, donde los primeros pasos realmente corresponden al proceso de entrenamiento, aunque son incluidos para clarificar la similitud de este algoritmo con el K-medias:





(a)



(b)

Figura 3.12: Núcleos de vecindad: “burbuja” (a) y “Gaussiano” (b).

1. Formar el MAO con  $K$  neuronas o clases.
2. Iniciar los vectores de referencia.
3. Para cada neurona  $\eta_i$ , calcular una copia de todos los datos que estén más cerca del vector de referencia  $w_i$ . En otras palabras, calcular la pertenencia de cada dato a la clase o región de Voronoi  $V_i$  asociada a la neurona  $\eta_i$ .
4. Actualizar cada vector de referencia  $w_i$  tomando la media del conjunto de muestras incluidas en las regiones de Voronoi asociadas a la neurona  $\eta_i$  y a sus vecinas (pertenecientes a  $\Xi_i$ ).
5. Reducir el radio de vecindad y repetir desde el paso 3 varias veces.

Formalmente, si se usa una función de vecindad  $h_{j,i}$  general y  $\bar{x}_j$  es la media de los datos  $x(t)$  en la región de Voronoi  $V_j$ , entonces ésta es ponderada por el número  $n_j$  de muestras en  $V_j$  y por la función de vecindad, obteniendo la siguiente regla de actualización:

$$w_i^* = \frac{\sum_j n_j h_{j,i} \bar{x}_j}{\sum_j n_j h_{j,i}} \quad (3.8)$$

donde la suma sobre  $j$  se toma para todas las neuronas del MAO si la función de vecindad no está truncada. En caso contrario, se toma para las neuronas en la vecindad  $\Xi_i$ . Por otro lado, si no se utiliza la ponderación por la vecindad se obtiene:

$$w_i^* = \frac{\sum_{j \in \Xi_i} n_j \bar{x}_j}{\sum_{j \in \Xi_i} n_j} \quad (3.9)$$

En relación al K-medias, que minimiza una función de energía (el error de cuantización, véase (2.2)), el mapa por lotes minimiza (aproximadamente) una función de energía que varía en el tiempo, dada la vecindad decreciente [Lut90]. En realidad, el algoritmo MAO no posee una función de energía en el caso general [EOS92]. En el caso de trabajar con un conjunto de datos discreto y un núcleo

de vecindad fijo la medida de distorsión:

$$E_{MAO} = \sum_{j=1}^N \sum_{i=1}^M h_{c,i} \|x_j - w_i\|^2 \quad (3.10)$$

es una función de energía local del MAO [KMSK91] (donde  $N$  es la cardinalidad del conjunto de datos,  $M$  la del conjunto de neuronas y  $c$  denota el índice de una ganadora). En cualquier caso, el MAO sólo da una solución aproximada a (3.10) dado que  $h_{c,i}$  decrece en cada iteración. Esta función es utilizada para comparar diversos algoritmos de EMD con los MAO.

#### 3.2.5. Proceso de entrenamiento

El proceso de entrenamiento de un MAO no sólo consiste en aplicar el algoritmo de entrenamiento con una serie de parámetros dados. Involucra la selección de la dimensión, topología y tamaño de la red, la elección una estrategia para inicializar los vectores de referencia y la ejecución del algoritmo de entrenamiento varias veces, además de la elección de los parámetros de entrenamiento en cada una de estas ejecuciones.

##### 3.2.5.1. Dimensión de la red

Al trabajar con grandes conjuntos de datos multidimensionales suele ser ventajoso descubrir o imponer alguna estructura sobre los datos. Una de estas estructuras es el número mínimo de parámetros  $d$  necesarios para describir los datos, también denominado “dimensión intrínseca” del conjunto de datos o del proceso generador de éstos. La interpretación geométrica de dimensión intrínseca es que el conjunto de datos se encuentra en una curva topológica de  $d$  dimensiones o menos [FO71]. En otras palabras, podemos decir que los datos “viven” en una hipersuperficie de dimensión  $d$ , empotrada en un espacio de dimensión  $n$ , con  $d \leq n$ . En la práctica, debido al ruido los datos podrán parecer de dimensión  $n$ , aunque seguirán pudiendo ser aproximados por una hipersuperficie de dimensión  $d$  [Kiv98]. Existen otras definiciones de dimensión intrínseca, por ejemplo, relacionadas con el ACP y correlaciones en los datos (número de componentes principales necesarios para representar los datos con precisión), véase [Bis95].

Considérese la Figura 3.13, en la que un MAO bidimensional se ha adaptado a una distribución con forma de superficie de una semiesfera. A pesar de que la

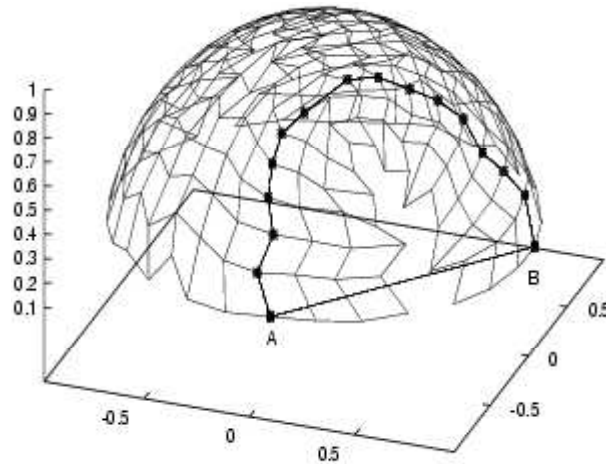


Figura 3.13: Adaptación de un MAO bidimensional a una distribución con forma de superficie de una semiesférica. Tomado de [BHV99].

distribución es definida en un espacio tridimensional, intuitivamente, parece más apropiado adaptar un plano (de neuronas o vectores de referencia) a la distribución que un cubo o un segmento. Para que un MAO se ajuste adecuadamente a una distribución de datos, su dimensión debe coincidir lo mejor posible con la dimensión intrínseca de los datos. Si la dimensión del MAO es menor que la intrínseca de los datos, éste intenta rellenar la hipersuperficie de los datos doblándose y distorsionándose de manera análoga a como una curva de Peano unidimensional rellena una superficie plana [Kiv98]. Esto introduce discontinuidades que pueden afectar negativamente al proceso de exploración de datos. En otras palabras, las neuronas en el mapa pueden quedar desordenadas. Por otro lado, un MAO de mayor dimensión que la intrínseca de los datos también presenta dificultades para adaptarse a la distribución. En principio, esto no resulta ser un problema serio ya que siempre es posible disminuir la dimensión del MAO. Por último, hay que destacar que una misma distribución de datos puede no presentar la misma dimensión intrínseca en todas sus regiones, por lo que los MAO pueden tener dificultades para adaptarse a estas distribuciones.

Una de las estrategias para estimar la dimensión intrínseca de los datos consiste en entrenar varios MAO y utilizar medidas de preservación de la topología, que sirven como medidas de bondad de MAO [GS96, BHV99] (véase la Sección 3.2.7). Se trata de entrenar varios MAO de diferente dimensión para posteriormente medir “lo bien” que se han adaptado a la distribución. Como un MAO de una dimensión dada se adapta mejor a datos de esa misma dimensión intrínseca,

los MAO que mejor se adapten a la distribución revelarán su posible dimensión intrínseca. Por ejemplo, mediante esta técnica se deduce que la dimensión intrínseca de los datos de habla utilizados en [BHV99] está comprendida entre dos y tres.

En la mayoría de aplicaciones y trabajos descritos en la literatura los MAO suelen ser bidimensionales. Una de las razones prácticas de su popularidad es que resultan sencillos de visualizar. Otra cuestión importante es que muchas implementaciones sólo soportan MAO de una o dos dimensiones [Kiv98]. Otra razón puede ser el “efecto de bordes” [Koh82] que es más pronunciado cuando la dimensión de los MAO aumenta.

#### 3.2.5.2. Topología y tamaño de la red

Otra de las etapas iniciales consiste en determinar la topología y el tamaño (número de neuronas) del MAO. Su elección depende principalmente del propósito para el que el MAO es aplicado: análisis de datos y/o clasificación. Una topología hexagonal resulta más efectiva para el análisis visual ya que no favorece tanto las direcciones (ejes) horizontal y vertical como la rectangular [KHKL96, Koh01]. En esta tesis se trabaja con rejillas rectangulares por los motivos mencionados en la Sección 3.2.3.3. En cualquier caso, la elección de la topología no parece crítica ni para tareas de visualización ni de clasificación.

En cuanto a la “forma” del MAO, se prefieren rectangulares en vez de cuadrados ya que la “red elástica” de vectores de referencia debe estar orientada según la densidad de probabilidad de la distribución de entrada y debe estabilizarse en el proceso de aprendizaje [Koh01]. En otras palabras, es deseable que los “ejes” de la rejilla correspondan a las direcciones de mayor varianza de los datos. Sin embargo, esta tesis utiliza principalmente MAO cuadrados ya que las distribuciones de entrada pueden ser “blanqueadas” (aplicando una transformación lineal de modo que la nueva matriz de covarianza sea la matriz unidad) para no favorecer ninguna dirección. Por otro lado, los mapas cuadrados también poseen dos direcciones “máximas”: las diagonales. En cualquier caso, la interpretación de los “ejes” del mapa casi siempre resulta difícil [Ves99].

Sin embargo, el tamaño de la red influye bastante a la hora de aplicar adecuadamente un MAO. Si el objetivo es construir un clasificador no es necesario modelar perfectamente la distribución de datos, sino crear una serie de fronteras de decisión apropiadas para clasificar lo mejor posible [Rip96a]. Esto puede

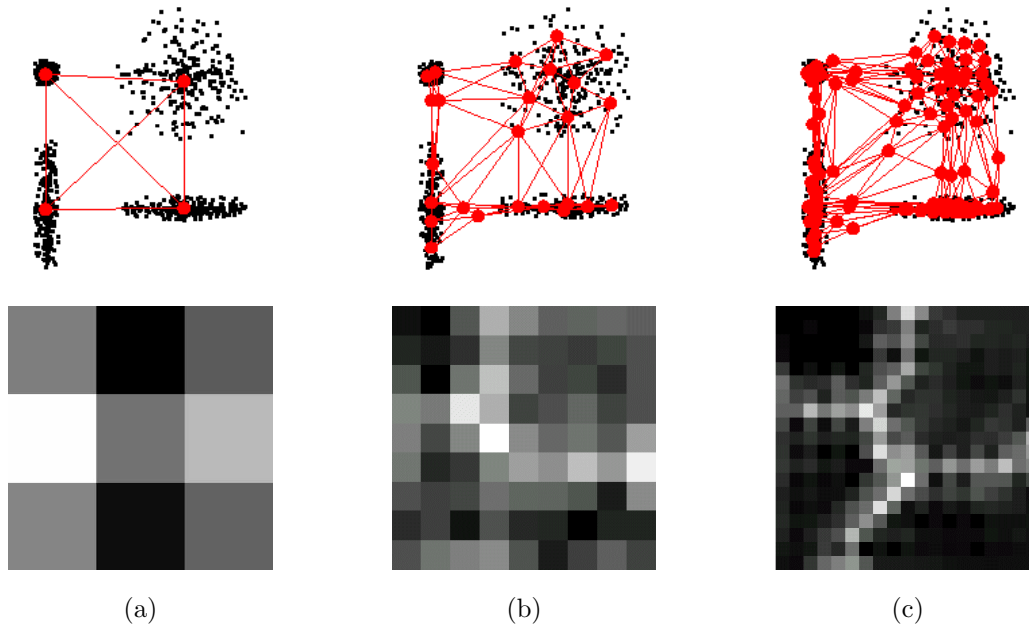


Figura 3.14: Adaptación de tres MAO de diferente tamaño a distribuciones similares formadas por cuatro *clusters* (imágenes superiores). Los métodos de visualización (que generan las imágenes inferiores) son más efectivos cuantas más neuronas sean empleadas.

conseguirse con un número reducido de neuronas y sus correspondientes vectores de referencia. En un extremo, cada neurona puede representar una clase de la misma forma que lo hace un centroide del algoritmo K-medias. Sin embargo, dado el proceso de adaptación de los MAO, suele ser muy difícil que cada vector de referencia se adapte a un *cluster* diferente (generalmente aparecen neuronas interpolantes, véase la Sección 3.2.6, que no se ajustan a ningún *cluster* y varias neuronas adaptadas a un mismo *cluster*), con lo cual es preferible emplear un número mayor de neuronas. El uso de un número elevado de neuronas para clasificación es aconsejable si al mismo tiempo se lleva a cabo un análisis visual del MAO para facilitar el etiquetado de éstas.

En cuanto a la visualización, el MAO podrá modelar mejor la distribución cuantas más neuronas tenga, por lo que es aconsejable utilizar un número elevado de ellas. Alfred Ultsch, autor del algoritmo más popular de análisis visual de MAO, señala que es necesario emplear miles de neuronas para crear MAO “emergentes”, y que el número de neuronas puede superar incluso al número de datos de la distribución [Ult99]. Por otro lado, un número elevado de neuronas aumenta el coste computacional de los algoritmos y puede provocar un “sobreajuste” del modelo a los datos [LK00].

La Figura 3.14 muestra, en las imágenes superiores, la adaptación de tres mapas bidimensionales con topología rectangular de diferentes tamaños:  $(2 \times 2)$  (a),  $(5 \times 5)$  (b) y  $(10 \times 10)$  (c), a unas simples distribuciones formadas por cuatro *clusters* de diferente forma pero con el mismo número de datos. Las imágenes inferiores muestran el resultado de aplicar el método de visualización U-matrix [US90] (véase la Sección 5.1.3) a los MAO.

En cuanto a la clasificación, el primer MAO, a pesar de contar con el mismo número de neuronas que de *clusters*, puede presentar errores de clasificación si se emplea la distancia Euclídea, ya que se supondría que las clases tienen forma esférica. Para reducir el error de clasificación es necesario emplear más neuronas, como en los casos (b) y (c). Posteriormente, el analista debe etiquetar las neuronas para determinar cuales representan a la misma clase. Los métodos de visualización pueden ayudar a realizar esta tarea al definir zonas que pueden representar a *clusters* y a bordes entre éstos.

En cuanto a la visualización, el MAO  $(2 \times 2)$ -dimensional no aporta información fiable sobre la distribución al ser demasiado pequeño. En general no es posible aplicar métodos de visualización con MAO tan pequeños. En el MAO  $(5 \times 5)$ -dimensional empiezan a aparecer zonas oscuras y claras que corresponden a *clusters* y a bordes entre éstos, respectivamente. Finalmente, el MAO  $(10 \times 10)$ -dimensional muestra fielmente las características de la distribución (existencia de cuatro *clusters* disjuntos, densidad relativa de los *clusters*, etc.). Este ejemplo puede parecer sencillo, ya que la distribución modelada es bidimensional. La utilidad real de los MAO y los métodos de visualización consiste en aplicarlos a distribuciones de dimensión elevada. El Capítulo 5 recoge una extensa colección de métodos de visualización, mientras el Capítulo 6 propone varios nuevos métodos de visualización alternativos.

#### 3.2.5.3. Iniciación de los vectores de referencia

Se trata de asignar unos valores iniciales a los componentes de los vectores de referencia. Existen diversas estrategias para realizar esta tarea:

- **Iniciación aleatoria.** Se ha demostrado que los valores pueden ser aleatorios de manera que los vectores de referencia aparezcan inicialmente desordenados en el mapa, ya que después de la fase de ordenamiento éstos acaban ordenados (véase la Sección 3.2.5.4). En cualquier caso, esto no significa que la iniciación aleatoria sea la mejor estrategia ni la más eficiente.

- **Iniciación en el dominio de los datos.** Esto es deseable aunque los valores iniciales sean escogidos aleatoriamente [Koh01]. Se trata de asignar valores iniciales escogiendo: los propios datos de la distribución (seleccionándolos aleatoriamente entre la población), o utilizando los vectores resultantes de una previa fase de CV.
- **Iniciación lineal.** Es posible ordenar los vectores de referencia iniciales según ciertas direcciones. Kohonen afirma haber obtenido buenos resultados utilizando los autovectores de la matriz de autocorrelación de los datos que correspondan a los autovalores mayores [Koh01]. De esta manera, se podría evitar la fase de ordenamiento.
- **Otras.** Una línea de investigación propuesta por el autor de esta tesis consiste en elaborar algoritmos para proyectar una serie de vectores de referencia preestablecidos a un mapa regular bidimensional, de manera que queden ordenados mediante la optimización de criterios de preservación de la topología, de bondad de MAO, y similares a los empleados en los métodos de EMD. Con estos nuevos mapas también podría evitarse la fase de ordenamiento.

#### **3.2.5.4. Fases del proceso de aprendizaje: ordenamiento, adaptación y convergencia**

Una vez elegidas las características del MAO y los vectores de referencia iniciales se procede a entrenar la red. Se trata de aplicar el algoritmo de entrenamiento repetidas veces con diferentes tasas de aprendizaje y radios de vecindad iniciales, que son escogidos cada vez más pequeños según se avanza en el entrenamiento, mientras que el número de iteraciones o muestras presentadas al algoritmo aumenta. Para un adecuado entrenamiento de los MAO el proceso de aprendizaje o entrenamiento puede ser dividido en tres fases diferentes [SM00]:

1. **Ordenamiento.** Esta fase consiste simplemente en ordenar los vectores de referencia en el espacio de los datos, por lo que es posible aplicar un método alternativo de ordenamiento como sustituto al algoritmo de aprendizaje de los MAO. Para que un MAO se ordene globalmente, el radio de vecindad debe ser grande al principio para posteriormente “encoger” con el tiempo. Se recomienda escoger un radio de vecindad inicial de por lo menos la mitad del diámetro de la rejilla de neuronas, ya que por el contrario podrían aparecer



diferentes zonas del mapa sensibilizadas a los mismos datos de entrada, vulnerando la propiedad de preservación de la topología. En cuanto a la tasa de aprendizaje inicial, también debe ser elevada (cercana a la unidad [Koh01]), para posteriormente decrecer de forma monótona. Por otro lado, el MAO se ordena muy rápidamente por lo que el número de muestras a procesar (iteraciones del algoritmo) puede ser bastante bajo, representando menos del 10 % del total de todo el proceso de aprendizaje.

2. Adaptación. Dados los elevados valores del radio de vecindad y tasa de aprendizaje iniciales, los vectores de referencia en la fase anterior se ordenan rápidamente pero también tienden a concentrarse ya que la fuerza de atracción es muy fuerte. Esto impide que el MAO se adapte adecuadamente a los datos. Por tanto, es necesario una nueva fase de entrenamiento con un radio de vecindad inicial relativamente pequeño y una tasa de aprendizaje inicial menor, que implica a su vez la utilización de un número elevado de iteraciones del algoritmo, por ejemplo, un 70 % del periodo de aprendizaje. En esta nueva fase, los vectores tienden a adaptarse a la distribución de manera ordenada (ya que parten de la configuración ordenada generada en la fase anterior), siempre que la dimensión intrínseca de los datos coincida con la del MAO.
3. Convergencia. Una vez alcanzada esta fase, el MAO ha debido adaptarse adecuadamente a la distribución. Sin embargo, al tratarse de un método de CV, es deseable minimizar en la medida de lo posible el error de cuantización (véase (2.2)). Éste puede ser reducido aplicando un algoritmo de CV como el K-medias, o el algoritmo de entrenamiento de los MAO una última vez. En este caso, el radio de vecindad inicial debe contener únicamente a la ganadora o a lo sumo unas pocas vecinas. La tasa de aprendizaje inicial suele ser pequeña requiriendo más iteraciones del algoritmo cuanto menor sea ésta. La precisión estadística, que es muy difícil conseguir cuando el espacio es de dimensión elevada dada la maldición de la dimensionalidad [Kas97], depende del número de iteraciones en esta fase [Koh01]. Por último, cabe destacar que el algoritmo MAO es equivalente al K-medias cuando el radio de vecindad afecta solamente a la neurona ganadora.

Con respecto al número total de iteraciones del proceso, Kohonen sugiere utilizar por lo menos 500 veces el número de neuronas en el MAO para conseguir

Fase	$\alpha(0)$	$\sigma(0)$	$T$
Ordenamiento	$[0.4, 0.8]$	$\sigma(0) > D/2$	$\approx 5\%$
Adaptación	$[0.05, 0.2]$	$\sigma(0) < D/4$	$\approx 70\%$
Convergencia	$[0.01, 0.05]$	$\sigma(0) < 2$	$\approx 25\%$

Tabla 3.1: Sugerencias para los parámetros del algoritmo de aprendizaje según la fase del proceso de aprendizaje.  $D$  representa el diámetro de la rejilla de neuronas del MAO.

precisión estadística [Koh01]. Por otro lado, si sólo unas pocas muestras están disponibles, éstas pueden ser recicladas hasta alcanzar el número deseado de iteraciones.

En función de la experiencia, la Tabla 3.1 propone una serie de valores aproximados, como mera sugerencia, para la tasa de aprendizaje inicial  $\alpha(0)$ , el radio de vecindad inicial  $\sigma(0)$ , y el número de iteraciones del algoritmo  $T$  en cada fase. En cualquier caso, el conjunto de valores óptimos para estos parámetros siempre será dependiente de la distribución de datos, desconocida *a priori*, con la que el MAO sea entrenado, por lo que su elección no es trivial. En la práctica, dado que el algoritmo de entrenamiento es computacionalmente muy ligero, se recomienda entrenar varios MAO con diferentes conjuntos de parámetros, e incluso varios con los mismos parámetros (dada la naturaleza aleatoria del algoritmo secuencial), para finalmente escoger uno de ellos, en función de medidas de bondad de MAO, que son descritas en la Sección 3.2.7. Por ejemplo, si el objetivo es realizar solamente una CV, es posible escoger el MAO con menor error de cuantización.

La Figura 3.15 muestra dos ejemplos de la configuración de los vectores de referencia al finalizar cada una de las fases del proceso de entrenamiento. En (a), un MAO unidimensional formado por 16 neuronas se adapta a una circunferencia cuya dimensión intrínseca es uno. En (b), un mapa  $(3 \times 3)$ -dimensional se adapta a una distribución bidimensional formada por cuatro *clusters*. En ambas imágenes se muestra de izquierda a derecha la configuración de los vectores de referencia tras completar la iniciación aleatoria de los vectores de referencia, el ordenamiento, la adaptación y la convergencia.

### 3.2.6. Ventajas y deficiencias

El objetivo de los MAO es crear una proyección topológicamente (es decir, localmente) ordenada de los datos en el sentido de una curva o superficie principal discreta. Además de los beneficios ofrecidos por la CV, los beneficios principales

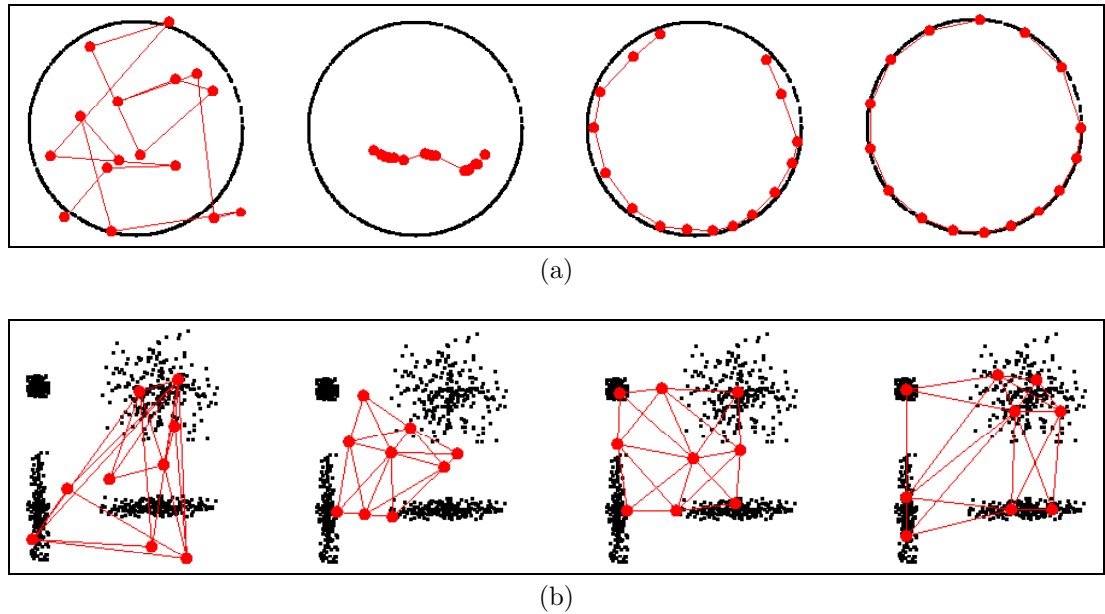


Figura 3.15: Fases del proceso de entrenamiento. De izquierda a derecha: previa iniciación de los vectores de referencia (aleatoria), ordenamiento, adaptación y convergencia. En (a), un MAO unidimensional formado por 16 neuronas se adapta a una circunferencia. En (b), un MAO  $(3 \times 3)$ -dimensional se adapta a una distribución bidimensional formada por cuatro *clusters*.

de los MAO son los siguientes [Ves00]:

- Robustez. Si el radio de vecindad inicial es suficientemente grande, abarcando la mayor parte del mapa, el MAO es muy robusto a la configuración inicial de los vectores de referencia.
- Adaptación local. El orden topológico trabaja principalmente en la vecindad de cada neurona, por lo que adapta los vectores de referencia localmente a la densidad de datos.
- Facilita la visualización. La rejilla regular del MAO permite crear visualizaciones e interfaces de usuario eficientes.

Algunos de los beneficios de los MAO son debidos a sus relaciones de vecindad, que por el contrario tiene efectos secundarios (véase la Figura 3.16):

- Efecto de bordes. La definición de la vecindad no es simétrica en los bordes del mapa, por lo que la estimación de la densidad es diferente en los bordes que en el centro del mapa.

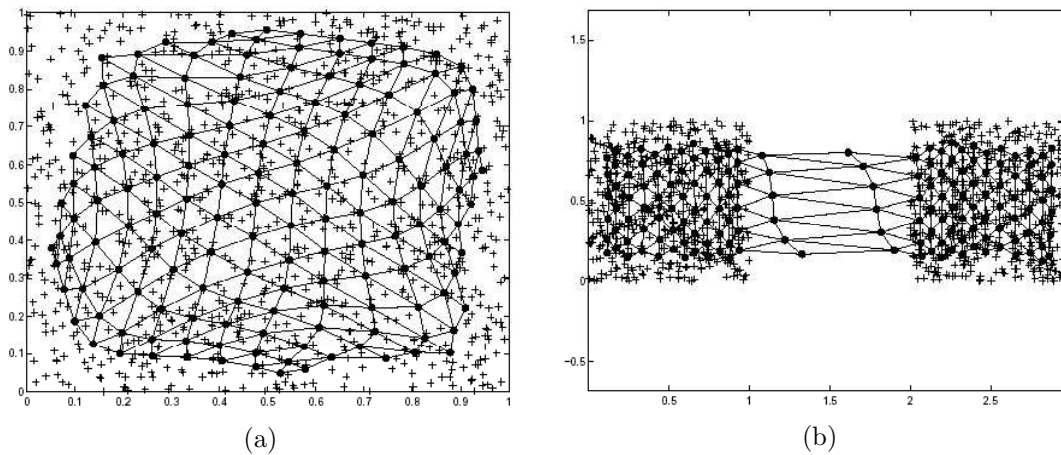


Figura 3.16: Algunos problemas asociados a los MAO: efecto de bordes y contracción (a), neuronas interpolantes (b). Tomados de [Ves02].

- Contracción. El rango de valores de las variables o componentes sufre una contracción. Los vectores de referencia de las neuronas exteriores son atraídos más hacia el centro del mapa, con lo que puede que no representen adecuadamente clases que pueden resultar interesantes desde el punto de vista del análisis, como son los datos atípicos.
- Neuronas interpolantes. Cuando existen *clusters* disjuntos, es habitual ver como ciertos vectores de referencia son adaptados a los espacios vacíos entre dichos *clusters*. Las neuronas correspondientes reciben el nombre de “interpolantes”. La aparición de éstas puede ser beneficioso desde el punto de vista de la visualización ya que representan bordes entre clases. Sin embargo, en el caso de algunas herramientas de análisis como el *clustering* de enlace simple, éstas pueden dar pistas falsas sobre la forma de los datos y puede ser necesario eliminarlas.

Por otro lado, la rejilla discontinua de los MAO presenta los siguientes problemas [Ves99]:

- Aunque el MAO tenga una forma rectangular, sus ejes raramente proporcionan una interpretación clara.
- La proyección discreta implementada por el MAO es muy básica. Los datos son proyectados sobre su neurona ganadora y suele ser muy complicado extraer información sobre la estructura global de la distribución utilizando

únicamente la rejilla. Por esta razón, es aconsejable combinar los MAO con otros métodos de proyección (por ejemplo, la proyección de Sammon).

El algoritmo del MAO es sencillo de implementar y simular, es muy ligero computacionalmente, y sus propiedades prácticas son claras y fáciles de observar. Sin embargo, sus propiedades teóricas siguen sin ser demostradas en un sentido general, a pesar del tremendo esfuerzo de varios autores. Sólo han aparecido avances teóricos con respecto a MAO unidimensionales, e incluso en este caso no se ha encontrado una tasa de aprendizaje conveniente que asegure la organización. Para MAO multidimensionales, el problema es aún más complicado [CFP94, CFP98]. Desde un punto de vista teórico, los MAO sufren varias deficiencias [BSW98, Koh01]:

- Ausencia de función de coste o energía a minimizar.
- Carencia de una base teórica para escoger una tasa de aprendizaje y núcleo de vecindad que asegure el ordenamiento topográfico, por lo que éste debe ser comprobado empíricamente.
- Ausencia de demostraciones generales de convergencia.
- El modelo no define una densidad de probabilidad.
- La propiedad de suavidad de los MAO viene determinada indirectamente por la función de vecindad, la cual cambia a través del entrenamiento por lo que es difícil de controlar.

Por otro lado, además del problema del “sobreajuste”, el simple uso del MAO puede llevar a un número excesivo de falsas hipótesis en tareas de visualización (“sobreinterpretación”) [LK00]. También hay que tener en cuenta que al combinar una CV con una proyección, los MAO pueden presentar limitaciones con respecto a otros métodos diseñados para realizar solamente una de estas tareas. Por ejemplo, el K-medias suele generar un conjunto de vectores de referencia con menor error de cuantización, y la proyección de Sammon preserva mejor las distancias entre los elementos que la rejilla regular rígida [Fle97]. De hecho, una alternativa ampliamente utilizada es aplicar un método de EMD sobre un conjunto de vectores de referencia provenientes de un método de CV. En cualquier caso, las aplicaciones de los MAO son cada vez más numerosas, especialmente en el análisis de datos [CFP94, CFP98].

### 3.2.7. Medidas de bondad

Dada la incertidumbre sobre la distribución de datos de entrada, al final del proceso de entrenamiento es conveniente verificar de alguna manera si el MAO generado se ha adaptado adecuadamente a los datos y si éste está ordenado, lo cual es un requisito primordial en cuanto a la validez y eficacia de la mayoría de métodos de visualización. Este apartado presenta consejos prácticos para construir “buenos” MAO, además de una serie de criterios denominados “medidas de bondad” de MAO. Algunas de estas medidas sirven como funciones de evaluación de algoritmos genéticos y estrategias evolutivas [Gol89, Mic92], para crear variantes optimizadas (híbridos de RNA–algoritmos evolutivos [Yao99, Gam01]) de los MAO [PU93, HH95, NH98].

#### 3.2.7.1. Consejos prácticos para la construcción de buenos MAO

Aunque es posible generar un buen MAO sin tomar ninguna precaución, Kohonen resalta una serie de consejos (algunos previamente mencionados en apartados anteriores) para crear MAO estables, bien orientados y menos ambiguos [Koh01]:

- Formar la rejilla. Para tareas de visualización se recomienda utilizar la topología hexagonal, mientras que la forma del mapa debe ser rectangular en vez de cuadrada. En las Secciones 3.2.3.3 y 3.2.5.2 se discute el uso en esta tesis de una topología rectangular y una forma cuadrada, respectivamente.
- Aprendizaje con un número pequeño de muestras. En muchas ocasiones, el número de iteraciones necesarias para obtener una buena precisión estadística supera al número de muestras. En estos casos, las muestras deben ser utilizadas reiterativamente en el aprendizaje. Es posible aplicarlas cíclicamente, en un orden permutado aleatorio o simplemente de manera aleatoria. Los resultados son prácticamente los mismos en los tres casos.
- Reforzar datos poco frecuentes. El MAO tiende a representar a la densidad de probabilidad de los datos  $p(x)$ , por lo que se ajusta a regiones donde existe una cantidad notable de datos. Sin embargo, en muchas aplicaciones, datos importantes aparecen con poca frecuencia estadística. En estos casos es posible reforzar el aprendizaje hacia estos datos, por ejemplo, aumentando el valor de la tasa de aprendizaje o del núcleo de vecindad cuando son

presentados, o introduciéndolos repetidas veces en el proceso de entrenamiento. La Sección 3.3.1.2 describe brevemente algunas de estas variantes.

- Escalado de los componentes de los patrones. Este es un problema complejo, ya que no existe una regla que determine el tipo de escalado óptimo que deba ser aplicado a los componentes de los datos al introducirlos en el algoritmo. En esta tesis los componentes son normalizados de modo que su varianza sea igual a uno.
- Forzar representaciones a lugares concretos del mapa. A veces es aconsejable localizar previamente el lugar en el mapa donde va a ser proyectado un cierto tipo de dato. Esto puede conseguirse inicializando el vector de referencia de la neurona en esa posición y manteniendo baja su tasa de aprendizaje.
- Monitorizar la calidad del aprendizaje. El aprendizaje puede ser llevado a cabo con diferentes parámetros de entrenamiento, diferentes conjuntos de vectores de referencia iniciales y diferentes secuencias de patrones de entrada. Parece claro que algún MAO óptimo puede existir de acuerdo con un criterio de bondad concreto. Al tratarse de un algoritmo de CV, un índice frecuentemente utilizado es el error de cuantización. Por otro lado, el MAO debe estar ordenado, presentando la propiedad de preservación de la topología que lo caracteriza. Este apartado describe varios criterios para medirla. Desafortunadamente, la preservación de la topología parece inversamente proporcional al error de cuantización, por lo que la elección de un MAO óptimo dependerá de la aplicación que se le quiera dar. Finalmente, cabe destacar que en la práctica varios MAO de diferente dimensión, tamaño y forma son entrenados con diferentes parámetros de entrenamiento, conjuntos iniciales de vectores de referencia, secuencias de entrada de patrones, etc., para finalmente escoger el óptimo según los criterios de bondad.

#### 3.2.7.2. Criterio de clasificación supervisada

Cuando un MAO es entrenado con el objetivo de crear un clasificador o reconocedor de formas, la bondad del mismo puede ser medida simplemente con la tasa de acierto del clasificador [VA95], suponiendo que posteriormente se dispone de información sobre la pertenencia de los datos a las distintas clases. Nótese que en este sentido lo importante es crear fronteras de decisión que separen adecuadamente las clases, en lugar de modelar fielmente la distribución de datos.

### 3.2.7.3. Bondad de la visualización

Para tareas de clasificación, es posible medir la separación entre las clases y la calidad del MAO objetivamente. Cuando el objetivo es análisis exploratorio, la calidad del MAO generalmente debe ser evaluada por un experto en el área de aplicación. En [KL96] se presenta una medida para comparar las imágenes generadas por dos MAO.

### 3.2.7.4. Error de cuantización medio

Algunos algoritmos de CV configuran sus vectores de referencia minimizando una función de coste o energía. Aunque el MAO no tiene una función de coste, es deseable que éste se adapte lo mejor posible a los datos. Después del proceso de entrenamiento casi siempre se realiza una estimación de dicha adaptación, por ejemplo, con medidas como el error de cuantización medio:

$$E = \frac{1}{N} \sum_{k=1}^K \sum_{x \in C_k} \|x - m_k\|^2 \quad (3.11)$$

donde  $N$  es el número total de muestras de la distribución,  $K$  es el número de clases y  $m_k$  es el centroide de la clase  $C_k$  (en los MAO naturalmente  $m_k = w_k$ ).

### 3.2.7.5. Medidas de preservación de la topología

A la hora de escoger un MAO óptimo no siempre debe ser elegido el que presente el menor error de cuantización medio. El MAO debe adaptarse a la distribución, pero de forma “ordenada”. Una extensa recopilación de medidas de organización es presentada en [Pol95]. Por tanto, también es necesario medir el grado de preservación de la topología que ha realizado el MAO, que suele ser “inversamente proporcional” al error de cuantización medio. Nótese que cuando el MAO converge a la distribución la estructura regular se distorsiona, “desordenándose” en el sentido de que pueden aparecer neuronas que representen datos similares, pero que se hallen lejos en el mapa. Por este motivo, es necesario buscar un equilibrio entre estos dos tipos de medidas.

Formalmente, la preservación de la topología puede definirse como el requisito de que la proyección del espacio de salida (de menor dimensión) al espacio de entrada original sea suave y continua [Kas97]. Sin embargo, es bastante difícil



de definir para una rejilla discreta. La mayoría de textos sobre MAO dicen, coloquialmente, que la preservación de la topología se satisface cuando los vectores de referencia de neuronas que se encuentran próximas en el mapa también se localizan próximos en el espacio de los datos (y no necesariamente a la inversa).

Existen dos enfoques principales para calcular la preservación de la topología. En el primero, se comparan las relaciones entre los vectores de referencia y las posiciones de sus neuronas asociadas en el mapa. En el segundo, se utilizan las muestras de la distribución para determinar la continuidad de la proyección desde el espacio de datos a la rejilla del mapa. En este sentido, si los dos vectores de referencia más próximos a un dato no son vecinos en el mapa, la proyección no es continua cerca de ese elemento.

A continuación se presentan varias medidas de preservación de la topología. Una introducción más extensa es descrita en [GS96] y [BHV99]. Cabe destacar que dada la ambigüedad de la definición de preservación de la topología existen diferencias importantes entre las medidas, por lo que no siempre generan resultados equivalentes para un mismo conjunto de datos y MAO.

- **Producto topográfico.** Esta medida  $P$  [BP92] relaciona para cada neurona la secuencia de vectores de referencia vecinos en el espacio de los datos con la secuencia de neuronas vecinas en el mapa. Sea  $d_V$  la distancia en el espacio de los datos ( $V$ ) y  $d_A$  la distancia en el espacio del mapa ( $A$ ). Para cada neurona  $\eta_j$ , se ordenan las distancias entre los vectores de referencia en el espacio de los datos y las distancias entre las neuronas en el mapa, de tal forma que  $n_i^V(j)$  denota el índice del  $i$ -ésimo vecino más cercano al vector de referencia de la neurona  $\eta_j$  en  $V$ , y  $n_i^A(j)$  el índice de la  $i$ -ésima neurona más cercana a la neurona  $\eta_j$  en  $A$ . El producto topográfico es definido mediante:

$$P = \frac{1}{N^2 - N} \sum_{j=1}^N \sum_{k=1}^{N-1} \log \left( \prod_{l=1}^k \frac{d_V(w_j, w_{n_l^A(j)})}{d_V(w_j, w_{n_l^V(j)})} \frac{d_A(j, n_l^A(j))}{d_A(j, n_l^V(j))} \right)^{\frac{1}{2k}} \quad (3.12)$$

El signo de  $P$  indica aproximadamente la relación entre la topología de ambos espacios. Si  $P < 0$  el espacio de los datos es de menor dimensión intrínseca,  $P \approx 0$  indica igualdad (aproximadamente) y  $P > 0$  indica que el espacio de datos es de mayor dimensión intrínseca.

- **Coefficiente de Spearman.** De forma similar al producto topográfico, en [BP95] se propone un criterio que preserva la clasificación del orden de

similitudes, en vez de las propias similitudes. Utilizan un coeficiente de correlación de clasificación entre las similitudes entre los dos espacios, en concreto el coeficiente  $\rho$  de Spearman (véase [PTVF93]), que es definido como el coeficiente de correlación lineal de las clasificaciones:

$$\rho = \frac{\sum_i (R_i - \bar{R})(S_i - \bar{S})}{\sqrt{\sum_i (R_i - \bar{R})^2} \sqrt{\sum_i (S_i - \bar{S})^2}} \quad (3.13)$$

donde  $R_i$  y  $S_i$  son las correspondientes clasificaciones en las listas ordenadas de las distancias en ambos espacios. El coeficiente  $\rho$  de Spearman tiene la propiedad deseable de que está acotado en el intervalo  $[-1, 1]$ . Un ordenamiento perfecto resulta si  $\rho = 1$ . Si el ordenamiento es aleatorio  $\rho \approx 0$ , y si está completamente invertido  $\rho = -1$ .

- Medida de Zrehen. Este enfoque [Zre93] considera el número de veces que el segmento conectando dos vectores de referencia vecinos en el mapa es cruzado por la region de Voronoi asociada a otro vector de referencia. En otras palabras, considérese dos vectores de referencia  $w_i$  y  $w_j$  cuyas neuronas son vecinas en el mapa. La esfera de radio  $\|w_i - w_j\|/2$  con centro en  $(w_i + w_j)/2$  no debe contener otros vectores de referencia, es decir:

$$\|w_i - w_k\|^2 + \|w_k - w_j\|^2 \leq \|w_i - w_j\|^2 \quad \forall k \neq i, j \quad (3.14)$$

A los  $w_k$  que violan esta condición se les denomina “intrusos”, y la suma del número de todos ellos sobre todos los pares de neuronas constituye la Medida de Zrehen. También es posible dividir esta medida por el número de pares de vectores de referencia que son vecinos en el espacio de datos [BHV99].

- Medida-C. Considere un espacio de entrada  $V$  y otro de salida  $A$ , los cuales contienen  $N$  puntos, y una proyección  $\Gamma$  del espacio  $V$  en  $A$ . Asuma que para cada espacio existe una función simétrica de similitud ( $F$  en el espacio de entrada y  $G$  en el de salida) que, dado cualquier par de puntos en el espacio, especifica mediante un número escalar no negativo lo similares que

son. La medida-C [GS96, GS97a, GS97b] es definida mediante:

$$C = \sum_{i=1}^N \sum_{j < i} F(i, j) G(\Gamma(i), \Gamma(j)) \quad (3.15)$$

donde  $i$  y  $j$  son puntos del espacio de entrada  $V$ , y  $\Gamma(i)$  y  $\Gamma(j)$  son sus respectivas imágenes en  $A$ . La suma se hace sobre todos los pares de  $V$ . Existen varias opciones a la hora de escoger las funciones  $F$  y  $G$ :

- Longitud de camino mínimo [DM90]:  $F(i, j)$  es la distancia Euclídea,  $G(p, q) = 1$  si  $p$  y  $q$  son vecinas, y  $G(p, q) = 0$  si no lo son.
- Cableado mínimo [DM90]:  $G(p, q)$  es la distancia Euclídea,  $F(i, j) = 1$  si  $i$  y  $j$  son vecinos, y  $F(i, j) = 0$  si no lo son.
- Distorsión mínima [Lut90]:  $F(i, j)$  es la distancia Euclídea al cuadrado.  $G(p, q) = e^{-d^2/\sigma^2}$ , donde  $d$  es la distancia Euclídea entre  $p$  y  $q$ , y  $\sigma$  es la escala de longitud en el espacio de salida sobre la que se desea que puntos cercanos representen puntos de entrada similares.
- Distorsión mínima invertida [Mit95]:  $G(p, q)$  es la distancia Euclídea al cuadrado.  $F(i, j) = e^{-d^2/\sigma^2}$ , donde  $d$  es la distancia Euclídea entre  $i$  y  $j$ , y  $\sigma$  es la escala de longitud en el espacio de entrada sobre la que se desea que puntos cercanos representen puntos de salida similares.
- Error topográfico. Esta simple medida [Kiv96] consiste en la proporción de datos cuyas neuronas asociadas a sus dos vectores de referencia más próximos en el espacio de los datos no son también vecinas en el mapa, es decir, donde aparece una discontinuidad local de la proyección:

$$\varepsilon_t = \frac{1}{N} \sum_{k=1}^N u(x_k) \quad (3.16)$$

donde  $u(x_k) = 0$  si las neuronas asociadas a los dos vectores de referencia más próximos a  $x_k$  son vecinas, y  $u(x_k) = 1$  si no lo son.  $N$  es la cardinalidad del conjunto de datos.

- Camino mínimo en el mapa. Esta medida [KL96] ( $\varphi$ ) tiene en cuenta tanto la preservación de la topología (un índice de la continuidad), como un criterio de ajuste estadístico a los datos (el error de cuantización). Es un intento de

combinar ambos tipos de medidas. Para cada dato de entrada  $x$  se calcula la distancia  $d(x)$  al segundo vector de referencia más próximo  $w_{c'(x)}$  a través del mapa, pasando primero por el más próximo  $w_{c(x)}$ , y a partir de ahí por el camino más corto siguiendo las conexiones (neuronas vecinas) del mapa hasta  $w_{c'(x)}$ . Si hay una discontinuidad en el MAO la distancia entre  $c(x)$  y  $c'(x)$  será elevada, mientras que si son vecinos será más corta. De esta manera, no sólo se cuantifica el error topográfico, sino que tiene en cuenta la magnitud de la discontinuidad. La medida  $\varphi$  es la media de estas distancias  $\varphi = E[d(x)]$ , donde:

$$d(x) = \|x - w_{c(x)}\| + \min_i \sum_{k=0}^{K_{c'(x),i}-1} \|w_{I_i(k)} - w_{I_i(k+1)}\| \quad (3.17)$$

donde  $I_i(k)$  es el índice de la unidad  $k$  en un camino en el mapa desde la unidad  $I_i(0) = c(x)$  hasta  $I_i(K_{c'(x),i}) = c'(x)$ . Las unidades  $I_i(k)$  y  $I_i(k+1)$  deben ser vecinas para  $k = 0, \dots, K_{c'(x),i} - 1$ . En [KL96] también se propone una medida de disimilitud entre MAO. La Figura 3.13 muestra un camino desde la neurona  $A$  hasta la  $B$  a través del MAO. La distancia es medida sobre la rejilla en el espacio de los datos, y no en línea recta.

- Medida Hebbiana. La rejilla de un MAO puede ser interpretada como un grafo. Por otro lado, es posible formar un grafo Hebbiano (también denominado “triangulación inducida de Delaunay”) conectando vectores de referencia según el principio de aprendizaje Hebbiano competitivo, que une dos de ellos si son los más próximos a un dato de entrada. La medida Hebbiana [Pol97] está basada en la comparación de estos grafos. También asigna unos pesos a las aristas del grafo Hebbiano para reducir la importancia de conexiones irrelevantes. Formalmente, dados dos vectores de referencia  $w_i$  y  $w_j$ , sea crea una arista Hebbiana  $c_{i,j} = n$ , donde  $n$  es un peso que corresponde al número de patrones de entrada que tienen a  $w_i$  y  $w_j$  como sus dos vectores de referencia más próximos. La medida Hebbiana  $\mu_H$  se define como:

$$\mu_H = 1 - \frac{\hat{c} \cdot |\mathcal{A}_K \setminus \mathcal{A}_H| + \sum_{(j,k) \in \mathcal{A}_K \setminus \mathcal{A}_H} c_{j,k}}{\hat{c} \cdot |\mathcal{A}_K| + \sum_{(j,k) \in \mathcal{A}_H} c_{j,k}} \quad (3.18)$$

donde  $\mathcal{A}_K$  denota el conjunto de aristas del grafo del MAO de Kohonen,  $\mathcal{A}_H$  el conjunto de aristas del grafo Hebbiano y  $\hat{c} = \sum_{(j,k) \in \mathcal{A}_H} c_{j,k} / |\mathcal{A}_H|$  es

la media de los pesos del grafo Hebbiano. Valores inmediatamente inferiores a uno indican un buen grado de preservación de la topología, mientras que valores positivos y cercanos a cero indican una pobre preservación de la topología.

- **Función topográfica.** En contraste con el producto topográfico, que utiliza las distancias entre los vectores de referencia en el espacio de los datos, la función topográfica  $\Theta$  [VDHM97] está basada en distancias  $d^D$  en el grafo inducido de Delaunay  $D$ . Las distancias entre dos puntos son calculadas en el grafo, y se compara el grafo del espacio de salida con el de entrada, con respecto al grado de desviación topológica entre ambos. Para cada par de vecinos en un espacio se cuenta el orden de vecindad del par correspondiente en el otro espacio. En concreto, para cada nodo  $i$  se evalúa la siguiente función:

$$f_i(k) = \text{card}\{j \mid \|i - j\|_{\text{máx}} > k; d^D(i, j) = 1\} \quad (3.19)$$

$$f_i(-k) = \text{card}\{j \mid \|i - j\| = 1; d^D(i, j) > k\} \quad (3.20)$$

donde  $k = 1, \dots, N-1$ , ( $N$  es la cardinalidad del conjunto de datos),  $\|\cdot\|$  es la distancia Euclídea,  $\|\cdot\|_{\text{máx}}$  la norma máxima, y  $\text{card}\{\cdot\}$  la cardinalidad de un conjunto. La función topográfica  $\Theta$  queda finalmente definida mediante:

$$\Theta(k) = \begin{cases} \frac{1}{N} \sum_{j \in A} f_j(k) & \text{si } k > 0 \\ \Theta(1) + \Theta(-1) & \text{si } k = 0 \\ \frac{1}{N} \sum_{j \in A} f_j(k) & \text{si } k < 0 \end{cases} \quad (3.21)$$

donde  $A$  es el espacio de salida. Esto genera  $\Theta \equiv 0$ , y en particular  $\Theta(0) = 0$  si tanto la proyección de  $V$  en  $A$ , como la de  $A$  en  $V$  preservan la topología. La forma de  $\Theta(k)$  permite analizar la magnitud de las distorsiones en el MAO, aunque si se quiere menos detalle es posible utilizar la diferencia:  $\bar{\Theta} = \Theta(+1) - \Theta(-1)$ . El signo de la función topográfica determina si la dimensión del espacio de entrada es demasiado elevada (negativo) o pequeña (positivo). Existe una relación estrecha entre la medida de Zrehen y  $\Theta(-1)$ , aunque no son exactamente iguales (la primera está basada en distancias Euclídeas, mientras  $\Theta$  lo está en la estructura de un grafo).

### 3.2.8. Relación entre el EMD y los MAO

La diferencia esencial entre las proyecciones generadas por los MAO y los métodos de EMD radica en que los primeros intentan crear una proyección localmente correcta, mientras que los segundos intentan preservar todas las distancias entre puntos. De esta manera, las proyecciones pueden ser muy diferentes. En el caso general, ningún método puede proyectar “perfectamente” datos de un espacio de dimensión elevada a otro de menor dimensión. En cuanto al AED, la diferencia puede ser interpretada de la siguiente manera: el orden global de los métodos de EMD probablemente será más preciso y proporcionará una visión más correcta de las distancias entre los datos. Por el contrario, el MAO intenta garantizar que los elementos proyectados en posiciones cercanas sean similares, y donde el orden y la estructura local de los *clusters* sea lo más fiable posible. La estructura global también puede ser visualizada pero se considera más importante la local.

En otras palabras, los métodos de EMD intentan preservar la métrica (o en el caso no-métrico, las relaciones globales de orden) del espacio original, mientras el MAO intenta preservar la topología, es decir, las relaciones de vecindad locales. La proyección de Sammon se encuentra en una posición intermedia, al preservar las estructuras locales y las distancias simultáneamente. La diferencia entre ambos tipos de métodos puede verse, por ejemplo, con la distribución semiesférica presentada en la Figura 3.13. Los métodos que preservan la topología necesitan dos dimensiones para representar los datos adecuadamente, mientras que los métodos que preservan las distancias necesitan tres.

Por otro lado, la complejidad de los métodos de EMD es  $\mathcal{O}(N^2)$ , donde  $N$  es el número de datos de la distribución. Esta complejidad es muy superior a la del MAO, que es  $\mathcal{O}(MT)$ , donde  $M$  es el número de neuronas del mapa y  $T$  el número de muestras presentadas al algoritmo.

En [KMJ95] se compara la proyección de Sammon con la del MAO. El criterio utilizado está basado en el error de Sammon (2.13), pero utiliza unas nuevas distancias para el MAO. De manera similar al método de preservación de la topología “camino mínimo en el mapa” (véase la Sección 3.2.7.5), es posible definir nuevas métricas en el espacio de observación o mapa. La distancia entre dos neuronas puede ser definida mediante la suma de las distancias (en el espacio original) a través del camino mínimo (en el grafo del mapa) entre los dos vectores de referencia asociados a las neuronas. Por otro lado, la distancia entre dos patrones en

el mapa puede ser definida mediante esta nueva distancia entre sus neuronas más próximas.

Finalmente, una práctica habitual es la combinación de ambos métodos [Him98, Fle01]. Es posible reducir la información a tratar por los métodos de EMD mediante la aplicación de técnicas de CV, para reducir el número de vectores a tratar. De esta manera, el algoritmo de EMD puede ser aplicado con los vectores de referencia de un MAO. Esto no sólo genera dos tipos de visualizaciones, sino que permite un encadenado de ambas para obtener información adicional, véase la Sección 5.5.2.

#### 3.2.9. Visualización, interpretación y uso

En cuanto a los métodos de visualización de información contenida en MAO la Sección 1.3 presenta una breve descripción de los más relevantes de la literatura. El Capítulo 5 lleva a cabo un análisis más detallado, mientras el Capítulo 6 propone diversos métodos nuevos, que son la principal aportación de esta tesis.

La interpretación de los descubrimientos del AED siempre depende de la aplicación, aunque algunas estrategias generales pueden ayudar al proceso de interpretación de las imágenes generadas por los métodos de visualización de MAO, véanse [Kas97] y los Capítulos 5 y 6.

La forma de la rejilla está predefinida y es regular por lo que cada neurona se sitúa en una posición única y todas tienen el mismo tamaño y forma en el mapa. Sin embargo, a pesar de la metáfora del mapa, el MAO transforma las posiciones de los datos de manera no-lineal, por lo que la escala de distancias no es la misma en todos los lugares del MAO, y no debe ser leído como un mapa clásico (por ejemplo, de carreteras). Por la misma razón, no parece aconsejable intentar interpretar los ejes vertical y horizontal del mapa. El MAO intenta sobre todo preservar las estructuras locales, por lo que su interpretación debe ser realizada localmente. En cualquier caso, la estructura global también puede ser útil.

Diferentes propiedades de los vectores de referencia pueden ser visualizadas en el mapa para facilitar su interpretación. Su organización “ordenada” ayuda a comprender las estructuras en los datos y a añadir (y visualizar) otro tipo de información. A medida que el analista queda familiarizado con el MAO, le es más fácil interpretar nueva información mostrada en él. En los MAO es posible detectar *clusters*, bordes entre éstos, sus densidades, sus relativos solapamientos y una medida orientativa de la distancia entre ellos, entre otras. Los diagramas

que muestran *clusters* son muy generales, en el sentido de que no debe asumirse la forma de éstos. También existen métodos para hallar *clusters* en los MAO automáticamente, véase [VA00].

Un problema frecuente a la hora de aplicar métodos estadísticos es cómo tratar datos incompletos, de los que se desconoce el valor de algunos (pocos) componentes. El MAO puede ser utilizado con vectores de datos incompletos si en la elección de la neurona ganadora sólo intervienen las variables conocidas. Si el número de componentes desconocidos es pequeño la comparación podrá ser estadísticamente bastante precisa. Al adaptar los vectores de referencia sólo los componentes disponibles en el dato serán actualizados.

Datos atípicos fruto del ruido o de otro tipo de errores apenas afectan al algoritmo de los MAO. Cada dato atípico afecta a una neurona, y el resto del mapa puede seguir siendo usando para inspeccionar los datos. Por otro lado, éstos pueden ser detectados fácilmente en los MAO, ya que por definición el espacio de datos es muy disperso cerca de los datos atípicos. En cualquier caso, por supuesto, pueden existir zonas dispersas que no correspondan a datos atípicos.

Es posible mostrar información en las propias neuronas para ayudar a la interpretación del MAO. Por ejemplo, los valores de los componentes de los vectores de referencia pueden ser visualizados mediante las técnicas comentadas en la Sección 2.5.2.2. Cuando los datos provienen de un proceso que varía en tiempo, es posible visualizar la trayectoria de las sucesivas neuronas ganadoras y monitorizar el estado del proceso. Por otro lado, información *a priori* sobre las clases a las que pertenecen los datos puede ser utilizada para etiquetar las neuronas, ayudando considerablemente a la interpretación de los MAO y a tareas relacionadas con la clasificación de datos de entrada. Existen también intentos de automatizar la interpretación del MAO [SVSV01].

Las imágenes generadas por los MAO sirven para obtener información acerca de un conjunto de datos. Han resultado ser eficaces a la hora de realizar AED, véanse [Ult93, MJ95, Kas97], y por tanto en tareas de minería de datos y descubrimiento de conocimiento con aplicaciones en, por ejemplo, análisis de textos, datos económicos y sistemas industriales [HKLK97, KK98, SVVH99]. A su vez, han sido utilizados con éxito en una amplia gama de tareas como el reconocimiento de patrones, análisis de habla, telecomunicaciones, análisis de imágenes, control de procesos, diagnóstico de fallos, robótica, diseño de circuitos electrónicos, IA, matemáticas, etc. Una extensa recopilación de aplicaciones de los MAO



aparece en [Koh01].

## 3.3. Variantes

### 3.3.1. Variantes clásicas de los MAO

El MAO básico tiene numerosas variantes. El factor común entre ellas es una colección de vectores de referencia y un conjunto de relaciones de vecindad entre ellos. Los vectores de referencia se ajustan a una distribución de datos y las relaciones son utilizadas de tal forma que los vectores de referencia vecinos resultan similares [Ves00]. Este apartado describe brevemente algunas de estas variantes.

#### 3.3.1.1. MAO de producto escalar

Los datos de entrada pueden ser normalizados (a longitud uno) antes de ser utilizados para entrenar un MAO. La normalización no es necesaria en principio, pero puede mejorar la precisión numérica ya que los vectores resultantes tienden a tener el mismo rango dinámico [Koh01]. Varios conjuntos de datos utilizados en esta tesis han sido normalizados siguiendo un criterio alternativo, que consiste en transformarlos para que todos los componentes tengan varianza uno.

Por otro lado, diferentes métricas pueden ser utilizadas a la hora de comparar o asociar vectores, aunque las reglas para identificar la neurona ganadora y para actualizar los vectores de referencia deben ser compatibles con respecto a la misma métrica. El primer algoritmo de entrenamiento de los MAO [Koh82] utilizaba el producto escalar como medida de similitud entre un dato  $x(t)$  y un vector de referencia  $w_i(t)$ . Esta variante parece tener una conexión con procesos fisiológicos [Koh01]. En este caso, las ecuaciones de aprendizaje quedan modificadas de la siguiente manera:

$$x^T(t)w_c(t) = \max_i \{x^T(t)w_i(t)\} \quad (3.22)$$

es utilizada para calcular la neurona ganadora  $\eta_c$  (donde  $T$  denota traspuesta), mientras la regla para actualizar los vectores de referencia es:

$$w_i(t+1) = \begin{cases} \frac{w_i(t) + \gamma(t)x(t)}{\|w_i(t) + \gamma(t)x(t)\|} & \text{si } i \in \Xi_c(t) \\ w_i(t) & \text{si } i \notin \Xi_c(t) \end{cases} \quad (3.23)$$

donde  $\Xi_c(t)$  es el conjunto de neuronas vecinas a la ganadora en un instante determinado,  $0 < \gamma(t) < \infty$  (Kohonen propone  $\gamma(t) = 100/t$  [Koh01]). Nótese que este proceso normaliza automáticamente los vectores de referencia.

### 3.3.1.2. MAO sensibles a datos de poca frecuencia

Uno de los consejos prácticos para construir MAO consiste en reforzar el aprendizaje hacia datos poco frecuentes. El MAO tiende de manera natural a adaptarse a la densidad de probabilidad de los datos  $p(x)$ . Esto es positivo en el sentido de que zonas de alta densidad de datos son representadas por un número mayor de vectores de referencia, aumentando la resolución del análisis visual en esas zonas. Los centroides tienden a representar a un número “similar” de datos, en comparación con otros algoritmos de CV, especialmente los que están basados en minimizar el error de cuantización como el K-medias. En otras palabras, la varianza del número de datos representados por cada centroide es menor. En esta tesis este tipo de algoritmos de CV son denominados “sensibles a la frecuencia”. Esta característica de los MAO también está presente en los mapas cerebrales (véase la Sección 3.2.1).

Sin embargo, en muchas aplicaciones, datos importantes aparecen con poca frecuencia estadística, y los MAO básicos no pueden modelar adecuadamente las clases a las que pertenecen. En estos casos el aprendizaje puede ser reforzado hacia dichos datos siguiendo diversas estrategias:

- Aumentar la tasa de aprendizaje para neuronas que ganan con poca frecuencia. En [AKCM90] se contabiliza el número de veces que una neurona ha sido la ganadora, disminuyendo la atracción de neuronas que han ganado más veces. Esto resulta equivalente a emplear una distancia modificada de manera que los datos aparenten estar más lejos de dichas neuronas.
- Modificar el núcleo de vecindad. En [SM96, SSM97] unas hiper-elipsoides que representan zonas de mayor densidad son definidas en el espacio de datos, y el núcleo de vecindad implementa una mayor atracción para las neuronas localizadas fuera de estas zonas. Otra idea similar consiste en utilizar la distancia de Mahalanobis para modificar el núcleo de vecindad [SM00].
- Utilizar una estructura jerárquica de MAO. Mediante esta estrategia es posible conseguir una resolución de acuerdo con la densidad de los datos

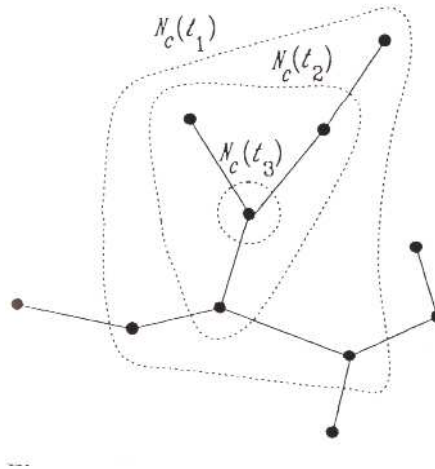


Figura 3.17: Vecindad del MAO de árbol generador mínimo. Tomado de [Koh01].

[SM00].

- Introducir más datos poco frecuentes en la secuencia de aprendizaje [Koh01].

### 3.3.1.3. MAO de árbol generador mínimo

En este tipo de MAO las relaciones entre las neuronas vecinas son definidas utilizando un árbol generado mínimo [KKL90]. Con respecto a la CV esta variante es más estable y mucho más rápida que el MAO básico. Sin embargo, los vectores de referencia no tienen una posición bien definida en la rejilla del mapa, por lo que la visualización es más problemática [Ves00]. La Figura 3.17 muestra la topología de la vecindad para este tipo de MAO.

### 3.3.1.4. MAO jerárquico

Con respecto a la minería de datos, la reducción de la complejidad computacional del algoritmo de los MAO es un aspecto tremendamente importante, dado el tamaño de las bases de datos a analizar. El MAO jerárquico [KO90, Koi95], también denominado “MAO estructurado en árbol”, es una variante especialmente eficiente de los MAO. Su estructura es dividida en una serie de niveles o capas que representan todo el espacio de datos. La diferencia entre las capas reside en que el número de vectores de referencia o neuronas crece exponencialmente a medida que se atraviesa un nivel. Las capas superiores son utilizadas para entrenar a las inferiores de manera que la velocidad de cálculo de la neurona ganadora es

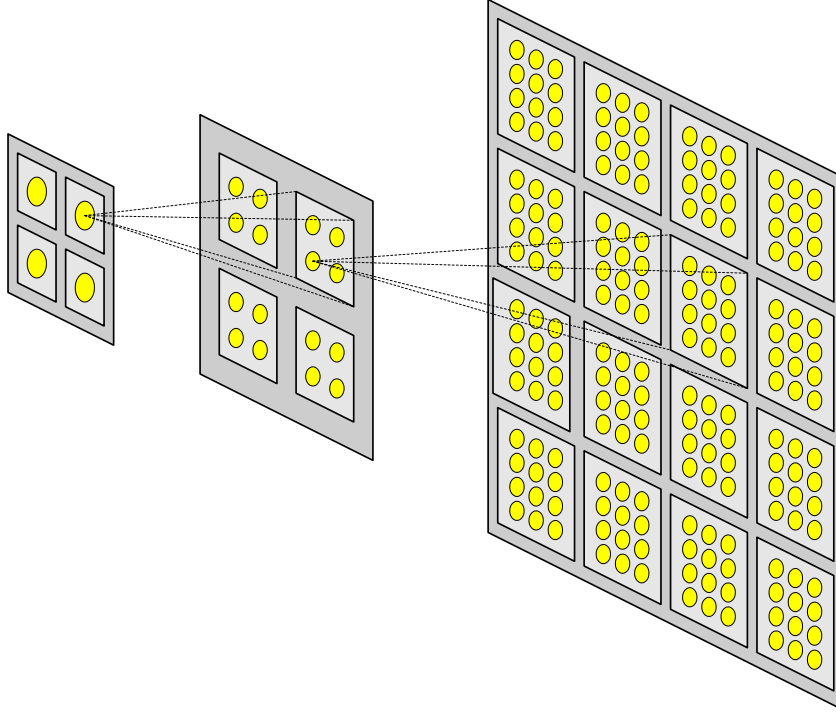


Figura 3.18: Estructura de un MAO jerárquico.

aumentada. Por ejemplo, si el número de neuronas  $M$  en cada capa  $c$  es expresada mediante  $M = \mu^c$ , donde  $\mu$  es el número de neuronas en el nivel inicial, la complejidad computacional del calculo de la ganadora en un determinado nivel pasa de  $\mathcal{O}(M)$  a  $\mathcal{O}(\mu \log_{\mu}(M))$ . En cualquier caso, no existe una norma estricta para determinar el número de neuronas en cada nivel, por lo que esto puede variar bastante. Desde el punto de vista de la visualización, a medida que se profundiza de nivel se obtiene una proyección mas detallada. La Figura 3.18 muestra la estructura arquitectónica de esta variante.

### 3.3.1.5. Redes con topologías flexibles

Algunos algoritmos relacionados con los MAO, y considerados variantes de éstos, intentan mejorar la preservación de la topología utilizando otras estructuras más flexibles para los mapas, diferentes de la rejilla regular fija [MS91, MBS93, MS94, Fri94, Fri95b, Fri96]. Destacan principalmente por sus buenas prestaciones en tareas de CV, sin embargo, no pueden ser utilizadas para visualización, al menos no tan fácilmente como las rejillas regulares [Kas97]. Esto es debido a que no definen un espacio de observación de menor dimensión diferente al de los datos,

que es donde se manifiesta la característica bioinspirada de auto-organización. Estos algoritmos proponen soluciones (siempre en el espacio de los datos) que definen vecindades de forma adaptativa, insertan y eliminan nodos de la red, y/o consideran el orden de clasificación de los vectores de referencia a las muestras, entre otras. Por otro lado, suelen ser algoritmos “sensibles a la frecuencia”, ya que los vectores de referencia tienden a representar a un número similar de patrones de entrada.

Por ejemplo, las redes de representación topológica [MS94], que consisten en la combinación del método de CV *neural gas* [MS91, MBS93], con el principio de aprendizaje Hebbiano competitivo [Mar93]. El método *neural gas* adapta una serie de vectores de referencia a una determinada distribución definiendo de manera adaptativa una vecindad entre éstos. Esta vecindad está basada en el orden de clasificación de las distancias entre dichos vectores y las muestras (primero, segundo, tercero, etc.), de manera que “aprenden” más los primeros que los últimos en función de su clasificación y no en su distancia real. Por otro lado, el principio de aprendizaje Hebbiano competitivo inserta una arista o conexión entre dos vectores de referencia si éstos son los dos (vecinos) más próximos a cualquier muestra de la distribución. El grafo resultante es un subgrafo de la triangulación de Delaunay que corresponde al conjunto de vectores de referencia [MS94]. Por otro lado, en [MS91] se implementa un procedimiento para eliminar aristas obsoletas, el cual está basado en un esquema de envejecimiento.

Las redes propuestas por Fritzke: *growing cell structures* [Fri94], *growing grid* [Fri95a] y *growing neural gas* [Fri96], introducen un elemento más en la adaptación al no utilizar un conjunto de vectores de referencia de tamaño fijo, sino que insertan y eliminan centroides en función de un criterio de error. Las vecindades son definidas cada vez que el conjunto de vectores de referencia es actualizado. El algoritmo *growing cell structures* es descrito brevemente en la Sección 2.5.5.5. El algoritmo *growing grid* es una variante (incremental) muy cercana a los MAO al insertar filas o columnas de centroides, con lo que la topología final se asemeja mucho a la de un MAO rectangular. El algoritmo *growing neural gas* puede ser visto como una variante del *growing cell structures* (en la que unas restricciones estrictas sobre la topología no son tenidas en cuenta) y como una variante incremental de las redes de representación topológica.

### 3.3.2. MAO secuenciales

Los MAO han sido diseñados para trabajar con patrones estáticos, sin embargo, éstos pueden ocurrir en secuencias, como las muestras espectrales obtenidas a partir de habla natural (correspondientes, por ejemplo, a 20 milisegundos de señal). Existen ciertas variantes secuenciales (también denominadas temporales o dinámicas) para tener en cuenta la dependencia estadística entre los patrones de entrada. Una técnica simple consiste en añadir información a la entrada de la red: concatenando varios patrones de entrada formando un “time-delay self-organizing map”, formando promedios exponenciales de varios patrones [Kan94], o utilizando una codificación en el dominio complejo, donde el tiempo es codificado en la fase o componente imaginaria [MADV95]. El uso de retroalimentación, similar a las redes recurrentes, nunca ha generado buenos resultados [Koh01]. Otra técnica consiste en utilizar una estructura jerárquica de dos MAO, donde el segundo intenta captar la dinámica espacial de las entradas moviéndose por el primero o introduce elementos para implementar memorias [Kan94, Car99, Voe00, VD01]. Existen métodos que integran memorias dentro del MAO, generalmente con trazas de memoria exponencialmente decrecientes [RP95, JM95, EP96, FP97]. Quizás las variantes secuenciales más populares son el mapa de Kohonen temporal [CT93], y el mapa auto-organizativo recurrente [VHM97], que son descritas brevemente en los siguientes subapartados.

Desde el punto de vista del autor de esta tesis, los MAO secuenciales han sido probados con secuencias muy simples, por lo que todavía queda por demostrar su utilidad a la hora de afrontar tareas complejas, como por ejemplo, el análisis de habla. En esta tesis los nuevos métodos de análisis visual han sido diseñados para ser utilizados con MAO básicos. Una línea futura consiste en analizar la capacidad de las variantes secuenciales y desarrollar métodos de visualización exclusivos para ellas.

#### 3.3.2.1. Mapa de Kohonen temporal

El mapa de Kohonen temporal [CT93] sólo difiere del MAO básico en sus salidas, es decir, al calcular la neurona ganadora. El MAO básico sólo tiene en cuenta la entrada a la red para calcular la neurona ganadora, que es equivalente a borrar toda la actividad de las salidas de las neuronas. El mapa de Kohonen temporal utiliza un mecanismo de memoria (“integrador con pérdidas”) para que una vez activada una neurona (con un determinado valor de salida), ésta pierda

su actividad gradualmente. Este modelo se asemeja al de las neuronas naturales, que retienen un potencial eléctrico en sus membranas con un decaimiento. Este decaimiento puede ser modelado mediante el siguiente potencial:

$$V_i(t) = dV_i(t-1) - \frac{1}{2}\|x(t) - w_i(t)\|^2 \quad (3.24)$$

donde  $0 < d < 1$ . La solución general a esta ecuación para una secuencia de entradas es:

$$V_i(t) = -\frac{1}{2} \sum_{k=0}^{n-1} d^k \|x(t-k) - w_i(t-k)\|^2 + d^n V_i(t-n) \quad (3.25)$$

La adaptación de los pesos en el instante  $t$  es análoga a la del MAO básico, salvo que la neurona ganadora es buscada de acuerdo con:

$$V_c(t) = \max_i \{V_i(t)\} \quad (3.26)$$

#### 3.3.2.2. Mapa auto-organizativo recurrente

El mapa auto-organizativo recurrente [VHM97] surge de mover las memorias (“integradores con pérdidas”) desde las neuronas de salida a las de entrada [KVHK97]. Se trata de utilizar la naturaleza temporal no sólo al buscar la ganadora, sino también a la hora actualizar los vectores de referencia. De esta manera, a cada neurona  $\eta_i$  se le asocia un vector de diferencias  $y_i(t)$  para un patrón de entrada  $x(t)$  en el instante  $t$ :

$$y_i(t) = (1 - \alpha)y_i(t-1) + \alpha(x(t) - w_i(t)) \quad (3.27)$$

donde  $0 < \alpha < 1$ . La solución general para (3.27) es:

$$y_i(t) = \alpha \sum_{k=0}^{n-1} (1 - \alpha)^k (x(t-k) - w_i(t-k)) + (1 - \alpha)^n y_i(t-n) \quad (3.28)$$

Ahora la neurona ganadora se busca mediante:

$$y_c = \min_i \{\|y_i(t)\|\} \quad (3.29)$$

Finalmente, la adaptación de los vectores de referencia está basada en los vectores de diferencias  $y_i(t)$ :

$$w_i(t+1) = w_i(t) + \gamma(t)h_{c,i}(t)y_i(t) \quad (3.30)$$

donde  $\gamma(t)$  es la tasa de aprendizaje y  $h_{c,i}$  el núcleo de vecindad.

### 3.3.3. MAO de subespacio adaptativo

En los MAO tradicionales cada neurona tiene asociado un vector de referencia que se actualiza en la dirección de un determinado vector de entrada y que representa a una clase de datos. Imagínese la situación si cada neurona tuviera dos vectores de referencia asociados. En ese caso, la neurona estaría definida, entre otros elementos, por un plano o base (excepto si los vectores son linealmente dependientes), que podría ser rotada en el entrenamiento para maximizar la proyección ortogonal de un nuevo dato de entrada sobre dicho plano. De esta manera, cada neurona representaría un subespacio lineal (adaptativo) y sería capaz de representar ciertos grupos de patrones elementales, invariantes con respecto a algunas transformaciones básicas como traslación, rotación y escalado. Esta es la idea subyacente de la variante mapa auto-organizativo de subespacio adaptativo cuya filosofía y algoritmo se describen en [Koh01].

El objetivo es utilizar un conjunto de características invariantes, de manera similar a las *wavelets* con respecto a la traslación. En el mapa auto-organizativo de subespacio adaptativo las unidades o neuronas desarrollan de forma adaptativa filtros de muchos tipos de características invariantes. Las formas matemáticas de estos filtros no necesitan ser descritas *a priori*, sino que éstas y sus mixturas encuentran su forma automáticamente en respuesta a transformaciones típicas que ocurren en las observaciones [Koh01].

La función más esencial del mapa auto-organizativo de subespacio adaptativo es el aprendizaje competitivo de episodios, los cuales representan secuencias de vectores de entrada, y a los cuales se adaptan las neuronas. Las bases de las neuronas, al igual que los episodios, pueden estar compuestas de varios vectores. Este esquema es único dentro de los modelos de RNA [Koh01]. Por otro lado, el mapa auto-organizativo de subespacio adaptativo difiere de otras RNA en el sentido de no aprender patrones, sino núcleos de transformación [Koh01]. Desde un punto de vista, este algoritmo está relacionado con las variantes secuenciales,



ya que trata secuencias de datos en el tiempo (episodios).

#### 3.3.4. Cuantización vectorial de aprendizaje

La cuantización vectorial de aprendizaje [Koh86, Koh01] es una variante supervisada de los MAO que representa una clase de algoritmos como el LVQ1, LVQ2, LVQ3 y OLVQ1. A diferencia de los MAO no definen una vecindad relaciona con la neurona ganadora, por lo que desaparece la organización espacial de los vectores de referencia. Por esta razón, este algoritmo es utilizado exclusivamente para clasificación estadística (supervisada), donde el objetivo es definir regiones asociadas a clases, que son representadas por los vectores de referencia.

A continuación se describe brevemente el LVQ1. Sea  $W = \{w_i(t)\}$  un conjunto de vectores de referencia donde cada  $w_i(t)$  representa a una clase predefinida  $C_i$  de un conjunto de datos, y sea  $x(t)$  un vector de entrada del que conocemos la clase a la que pertenece. El algoritmo es similar al del MAO, salvo que la adaptación se realiza solamente sobre la neurona ganadora mediante la siguiente regla:

$$w_c(t+1) = \begin{cases} w_c(t) + \alpha(t)[x(t) - w_c(t)] & \text{si } x \in C_c \\ w_c(t) - \alpha(t)[x(t) - w_c(t)] & \text{si } x \notin C_c \end{cases} \quad (3.31)$$

Es decir, dado un vector de entrada  $x(t)$ , la neurona ganadora se mueve hacia él si ambos pertenecen a la misma clase, o se aleja en caso contrario. La tasa de aprendizaje  $\alpha(t)$  es una función “lentamente” decreciente (su valor inicial suele ser pequeño, por ejemplo, 0,1) [Koh01].

# Capítulo 4

## BASES DE DATOS

Los métodos de visualización de MAO tratan de generar imágenes que revelen información útil acerca de la distribución de datos con la que han sido entrenados. El Capítulo 5 reúne una extensa recopilación de estos métodos, mientras el Capítulo 6 propone varios métodos nuevos de visualización, que son la principal aportación de esta tesis. Para ilustrar, evaluar y comparar los métodos es necesario considerar una serie de distribuciones de datos. El estudio llevado a cabo en esta tesis está basado en el análisis de MAO bidimensionales, entrenados con dos tipos de distribuciones de datos:

1. Distribuciones de simulación.
2. Datos reales de habla.

Por otro lado, algunos ejemplos de los métodos de visualización de MAO están tomados directamente de trabajos de la literatura, los cuales utilizan otros conjuntos de datos.

### 4.1. Distribuciones de Simulación

Las distribuciones de datos artificiales o de simulación sirven para poder evaluar de forma general las propiedades de los diferentes métodos de visualización de MAO. En contraste con las distribuciones de datos reales, las distribuciones de simulación son aconsejables por varias razones:

- Pueden ser diseñadas de manera que reúnan determinadas características, conocidas *a priori*, que posteriormente queramos que sean detectables por

los métodos de visualización de MAO. De esta manera, resultan adecuadas para evaluar si los métodos de visualización son capaces de representar dichas características, permitiendo establecer con claridad las propiedades de los diversos algoritmos de visualización.

- Los resultados son más objetivos y no es necesario un profundo conocimiento de los datos a analizar por parte del analista o experto.
- No es fácil encontrar datos reales que reúnan características especiales que queramos detectar.
- Es posible controlar factores como el nivel de ruido, datos atípicos, grado de solapamiento entre las clases, etc.

Las distribuciones de simulación han sido diseñadas principalmente para evaluar y comparar los diferentes algoritmos de visualización de MAO. Éstas están definidas en espacios de distinta dimensión ( $\mathbb{R}^2, \mathbb{R}^3, \mathbb{R}^8$ ), y su dimensión intrínseca puede ser 1, 2, 3 ó 8. La mayoría de distribuciones de simulación están compuestas por 2000 datos, excepto las dos primeras bidimensionales y otras 8-dimensionales formadas por *clusters* disjuntos. La mayoría de distribuciones pueden ser analizadas adecuadamente por otros métodos de AED, y varias de ellas resultan especialmente difíciles de tratar por algoritmos de clasificación automática o *clustering*. El tratamiento de distribuciones cuya dimensión intrínseca no coincide con la del MAO también es interesante ya que esto suele ocurrir en la práctica, donde aparecen los efectos de desorganización de las neuronas típicos de esta situación. La proyección de los MAO bidimensionales no puede preservar perfectamente la topología, lo que provoca problemas a la hora de visualizar la distribución, al aparecer discontinuidades en los MAO (véase la Sección 3.2.5.1). Los conjuntos de datos de simulación son descritos a continuación.

##### 4.1.1. Distribuciones bidimensionales

Los MAO utilizados en esta tesis son bidimensionales por motivos de visualización. Éstos suelen adaptarse ordenadamente a distribuciones cuya dimensión intrínseca es dos y suelen tener dificultades en caso contrario. Este apartado describe varios conjuntos de datos bidimensionales de dimensión intrínseca uno o dos. Aunque es posible definir estas distribuciones en espacios de dimensión elevada, en este primer apartado son definidas en el plano  $\mathbb{R}^2$ .

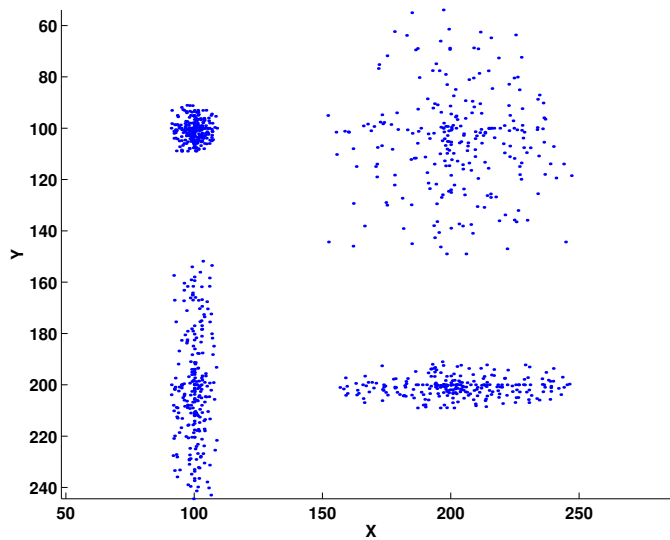


Figura 4.1: Distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

#### 4.1.1.1. Cuatro *clusters* de igual tamaño y diferente forma

La primera distribución (véase la Figura 4.1) consta de cuatro *clusters* disjuntos, con el mismo tamaño o número de elementos (250), pero con matrices de covarianza (formas) diferentes. Con este conjunto de datos es posible ver como el número de vectores de referencia que se adaptan a un *cluster* no sólo depende del número de datos pertenecientes a éste, sino también de su distancia al resto de *clusters* y de su densidad o error de cuantización medio.

#### 4.1.1.2. Círculo y anillo

La Figura 4.2 muestra una distribución formada por un círculo y un anillo de un determinado grosor concéntricos (ambas distribuciones son no-uniformes, donde existe mayor cantidad de datos en dirección al origen de coordenadas). El área que corresponde a cada clase es la misma ( $\pi$ ) y ambas poseen la misma cantidad de patrones. Se trata de una distribución relativamente difícil de tratar por métodos de clasificación convencionales (ambas clases tienen la misma media y es necesario definir varios vectores de referencia para modelar el anillo adecuadamente).

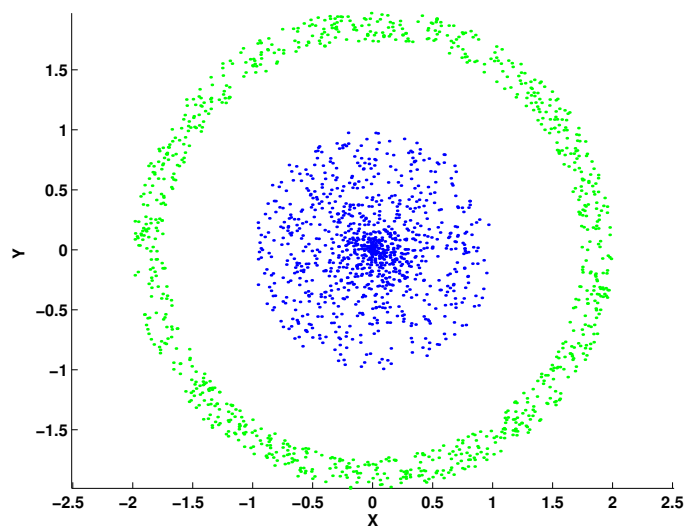


Figura 4.2: Distribución formada por un círculo y un anillo grueso concéntricos.

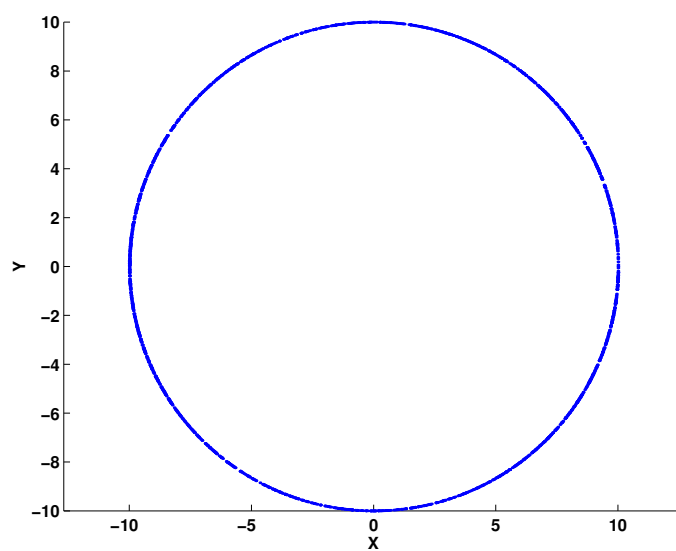


Figura 4.3: Distribución formada por una circunferencia.

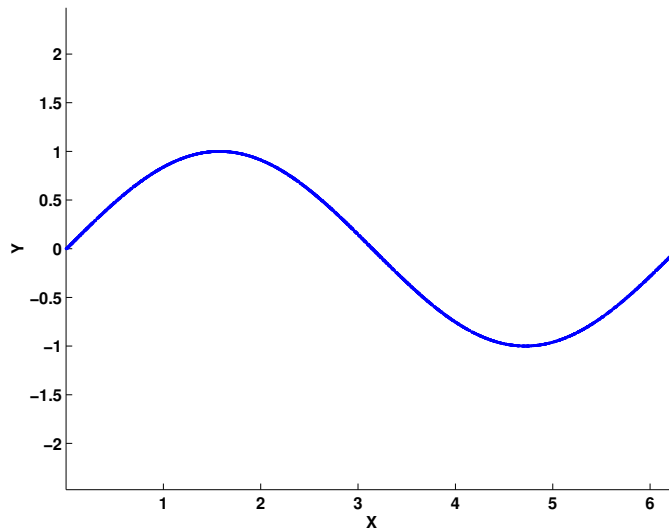


Figura 4.4: Distribución formada por un periodo de una senoide.

#### 4.1.1.3. Circunferencia

Las distribuciones de dimensión intrínseca uno también suelen ser difíciles de tratar en general ya que se alejan del concepto de *cluster*, donde se supone que los datos están ubicados alrededor de un centroide. En el caso de la circunferencia, donde la distribución es uniforme (véase la Figura 4.3), aparecen los fenómenos de desorganización típicos de una pobre preservación de la topología (especialmente si el MAO contiene un elevado número de neuronas), ya que los vectores de referencia organizados en una rejilla plana compiten por adaptarse a una curva “unidimensional”. El efecto es como si comprimiéramos un folio de papel en una vara delgada. En cualquier caso, suele ser posible detectar la estructura cíclica de la distribución con métodos clásicos de visualización de MAO.

#### 4.1.1.4. Senoide

La distribución uniforme formada por un periodo de una senoide (véase la Figura 4.4) es otro caso donde no existe una estructura de *clusters*. Tiene los mismos problemas que la circunferencia y su forma sinusoidal también suele poder ser visualizada adecuadamente con métodos tradicionales de visualización de MAO.

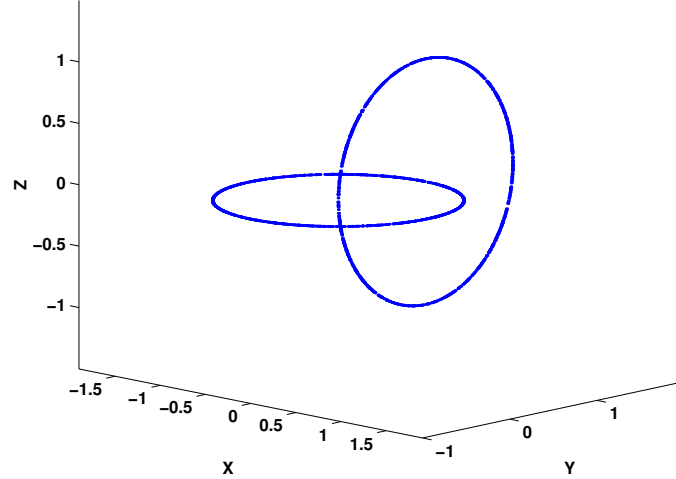


Figura 4.5: Distribución formada por anillos o eslabones encadenados.

### 4.1.2. Distribuciones tridimensionales

Las distribuciones presentadas en este apartado han sido definidas en  $\mathbb{R}^3$ , por lo que es posible simular situaciones en las que la dimensión intrínseca de los datos (tres) es superior a la dimensión del propio MAO. Este apartado también describe distribuciones cuya dimensión intrínseca es uno o dos.

#### 4.1.2.1. Dos eslabones encadenados

Esta distribución, ilustrada en la Figura 4.5, está formada por dos circunferencias (uniformes) encadenadas. Ha sido utilizada en la literatura para ilustrar el método U-matrix [UV94, Ult99], véase la Sección 5.1.3. Resulta muy difícil de tratar por métodos clásicos de clasificación, aunque los MAO pueden detectar la presencia de las dos clases mediante técnicas de visualización o incluso automáticamente. Sin embargo, los métodos de visualización tradicionales no detectan la estructura cíclica de las clases, la cual puede ser vista mediante varios métodos propuestos en esta tesis.

#### 4.1.2.2. Dos superficies planas paralelas

La Figura 4.6 muestra una distribución formada por dos superficies planas paralelas donde la distribución es uniforme. La dimensión intrínseca de los datos es dos, por lo que un MAO bidimensional no debe tener problemas para adaptarse a

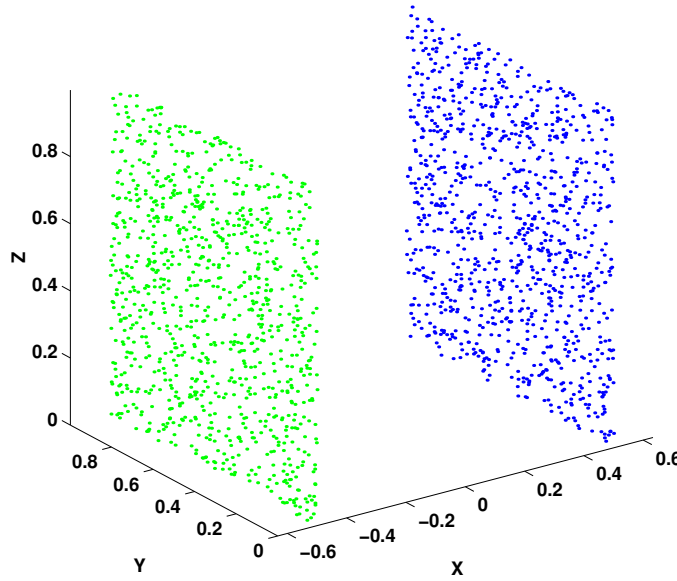


Figura 4.6: Distribución formada dos superficies planas paralelas.

ellos de forma ordenada. Sin embargo, dada la posición relativa de las superficies, resulta difícil preservar la topología en todas las zonas del MAO, apareciendo discontinuidades que no pueden revelar los métodos clásicos de visualización. En cualquier caso, éstos son capaces de discriminar las dos clases correspondientes a cada superficie. La Sección 6.4 propone varios métodos para analizar las discontinuidades de un MAO, los cuales están basados en la proyección de grafos sobre el espacio de observación de un MAO.

#### 4.1.2.3. Estrella

La distribución presentada en la Figura 4.7 corresponde a tres ortoedros delgados (donde la distribución es uniforme) dispuestos según las direcciones de los ejes de coordenadas. Aunque los MAO deben ser analizados especialmente de forma local, este ejemplo describe una distribución cuya estructura global puede ser visualizada aceptablemente con MAO bidimensionales, mientras que la local es difícil de interpretar, dada la dimensión intrínseca de los datos.

#### 4.1.2.4. Esfera maciza dentro de una esfera hueca

La distribución presentada en este apartado resulta muy difícil de analizar por MAO bidimensionales ya que la dimensión intrínseca de los datos es claramente



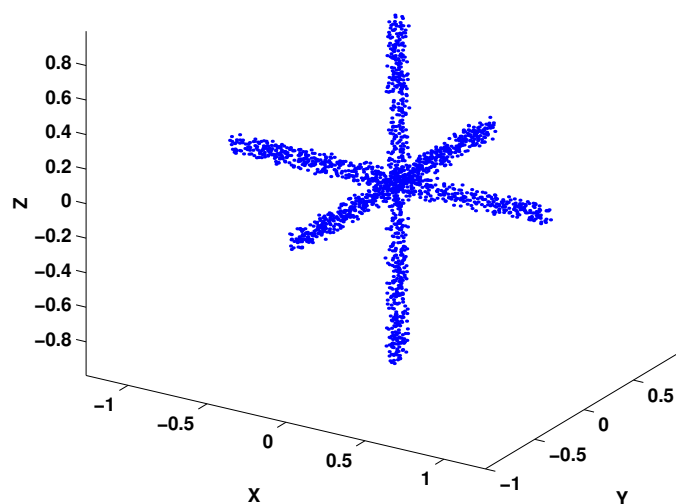


Figura 4.7: Distribución con forma de estrella.

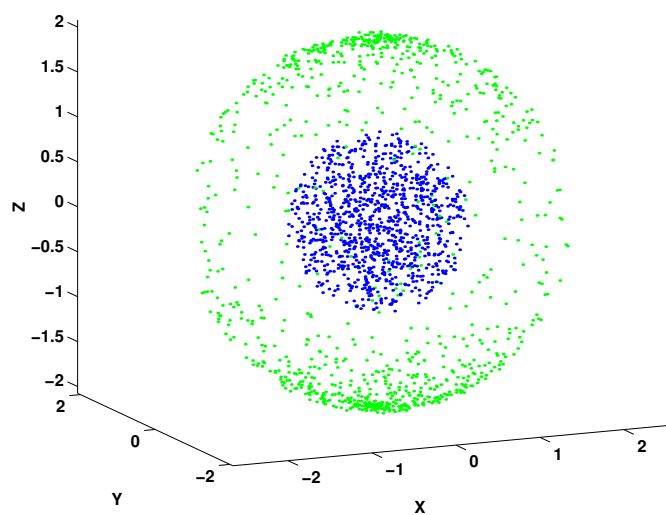


Figura 4.8: Distribución formada por una esfera maciza recubierta por otra hueca.

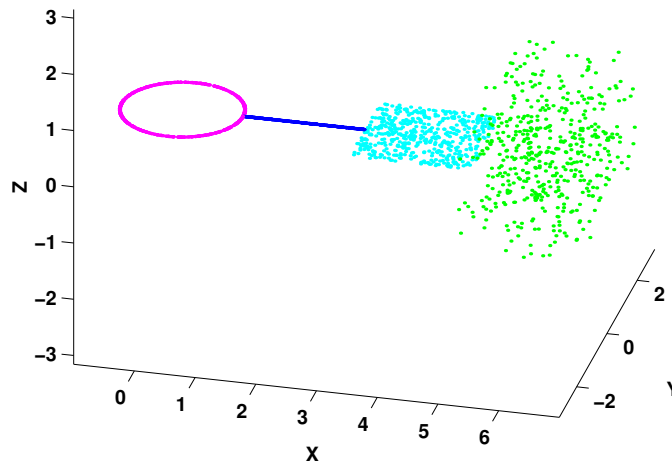


Figura 4.9: Distribución formada por una circunferencia, un segmento, una superficie cuadrada y un ortoedro.

tres. Se trata de una distribución formada por una esfera maciza (uniforme) de radio uno, rodeada por otra hueca (no-uniforme) con un ligero grosor, véase la Figura 4.8. Ambas distribuciones encierran el mismo volumen  $4\pi/3$ . En general, los métodos de análisis visual de MAO no pueden representar adecuadamente las propiedades de la distribución, sin embargo, pueden llegar a discriminar ambas clases.

#### 4.1.2.5. Circunferencia, segmento, cuadrado y ortoedro

La Figura 4.9 presenta otro ejemplo utilizado en la literatura (véanse [MS94, Fri95b, Fri96]). Se trata de una concatenación de una circunferencia, un segmento, una superficie plana cuadrada y un ortoedro. En todos los casos la distribución es uniforme. Resulta un ejemplo muy interesante ya que está compuesto de datos de menor, igual y mayor dimensión intrínseca que la dimensión de un MAO bidimensional, acercándose a una posible situación real.

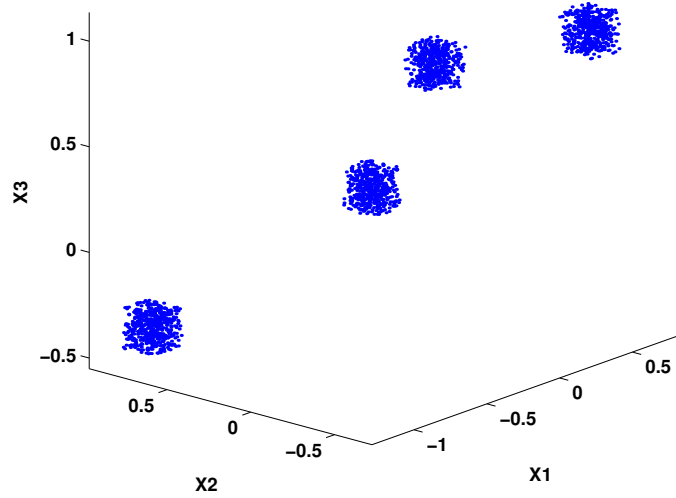


Figura 4.10: Distribución formada por cuatro *clusters* disjuntos con forma hipercúbica.

### 4.1.3. Distribuciones multidimensionales

En este apartado las distribuciones de datos han sido definidas en  $\mathbb{R}^8$ . Las tres primeras están formadas por varios *clusters* disjuntos de dimensión intrínseca ocho, mientras la dimensión intrínseca de las dos últimas distribuciones es igual a uno (a efectos de evaluar los métodos de visualización estas últimas podrían haber sido definidas igualmente en los apartados anteriores).

#### 4.1.3.1. Cuatro *clusters* disjuntos con forma hipercúbica

La distribución de la Figura 4.10 muestra una proyección de cuatro *clusters* disjuntos con forma hipercúbica sobre el espacio definido por sus tres primeras componentes. La distribución contiene 2000 datos y es uniforme en los *clusters*. Con este sencillo ejemplo es posible analizar las características generales de los métodos de visualización de MAO al trabajar con distribuciones de dimensión elevada.

#### 4.1.3.2. Ocho *clusters* disjuntos con forma hipercúbica

La Figura 4.11 muestra básicamente el mismo ejemplo que el anterior, salvo que el número de *clusters* es incrementado hasta ocho y el número de muestras

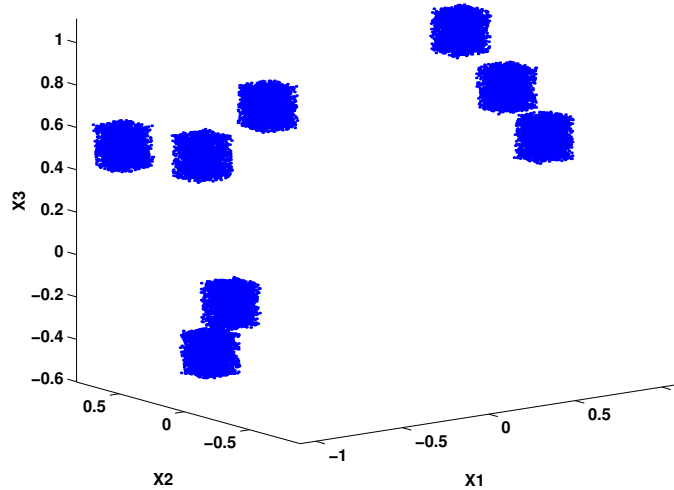


Figura 4.11: Distribución formada por ocho *clusters* disjuntos con forma hipercúbica.

hasta 20000. Comparando las imágenes de los métodos de visualización de MAO para las distribuciones de cuatro y ocho *clusters* es posible apreciar que existe un límite en el número de *clusters* que pueden ser detectados con un MAO de un tamaño dado.

#### 4.1.3.3. Cuatro *clusters* disjuntos de diferente tamaño e igual forma

Esta distribución (véase la Figura 4.12) consta de cuatro *clusters* disjuntos, con forma hipercúbica y distribución uniforme, compuestos por un número diferente de elementos: 8000, 2000, 4000 y 6000, en los *clusters* coloreados verde, rosa (*magenta*), azul claro (*cyan*) y azul, respectivamente. Con esta distribución se persigue principalmente averiguar si un método de visualización es capaz de mostrar todos los bordes entre los *clusters*, es decir, discriminar las cuatro clases (obsérvese la pequeña distancia relativa entre las parejas de *clusters* centrados en  $X1 = -10$  y en  $X1 = 10$ ).

#### 4.1.3.4. Un segmento

Un caso particular de distribución de dimensión intrínseca uno es el segmento (véase la Figura 4.13). Su estructura lineal resulta muy difícil de visualizar con métodos clásicos de análisis de MAO. En el ejemplo, la distribución en el segmento es uniforme y sólo son ilustradas sus tres primeras componentes.

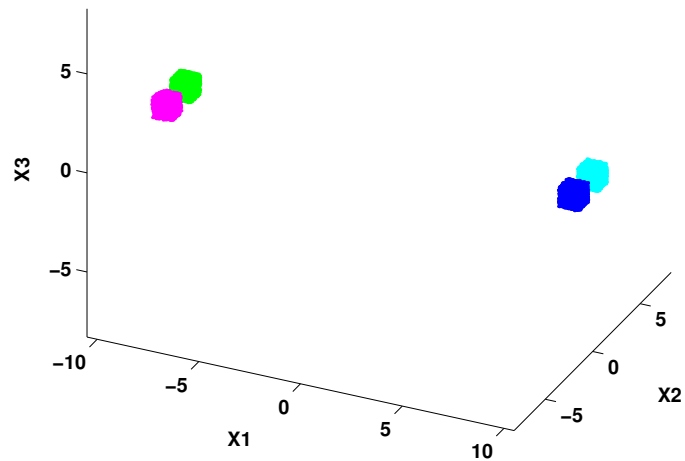


Figura 4.12: Distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma.

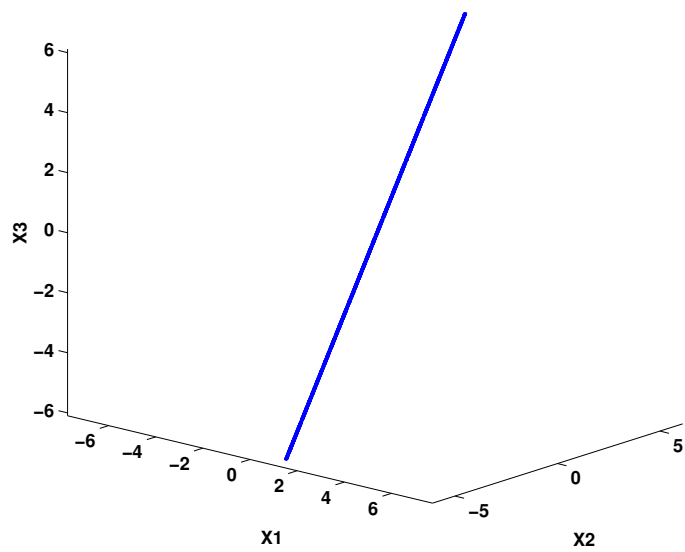


Figura 4.13: Distribución formada por un segmento.

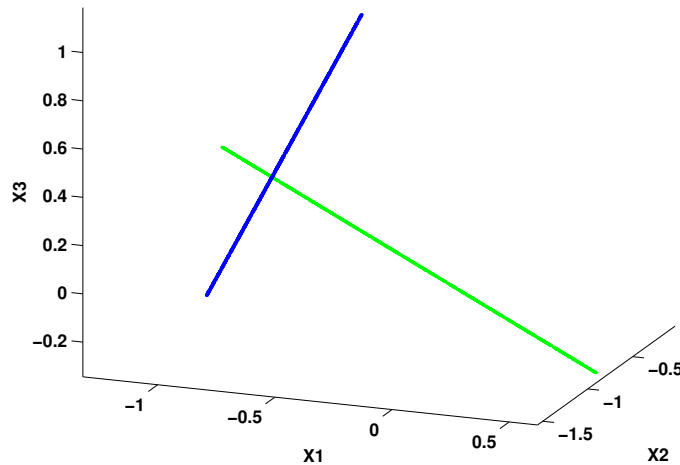


Figura 4.14: Distribución formada por dos segmentos.

#### 4.1.3.5. Dos segmentos

La última distribución de simulación, presentada en la Figura 4.14, está compuesta por dos segmentos (uniformes), de los cuales se ilustran únicamente sus primeras tres componentes. Al igual que en el caso anterior, resulta muy difícil determinar la estructura lineal de las dos clases, aunque los métodos de visualización de MAO las discriminan fácilmente.

## 4.2. Datos Reales de Habla

Una vez examinadas las propiedades de los diversos algoritmos de visualización (capacidad para detectar *clusters*, bordes entre *clusters*, datos atípicos, relaciones entre variables, nuevas estructuras, etc.), es tarea del analista aplicarlos adecuadamente a conjuntos de datos reales, en busca de ciertas características de los datos o para descubrir nueva información sobre las distribuciones. En comparación con los MAO entrenados con datos de simulación (artificiales), destacan las siguientes características de los MAO entrenados con datos reales:

- La verdadera utilidad de los MAO reside en la visualización de conjuntos reales de datos de dimensión elevada.

Locuciones de “right”	Hombres	Mujeres
Normal	20	20
Ruido inducido	20	20

Tabla 4.1: Distribución de locuciones de la palabra “right”.

- Las imágenes generadas a partir de datos reales son más difíciles de interpretar, al influir factores como el ruido, casos atípicos, variabilidad o solapamiento de los datos, etc.
- Normalmente las conclusiones no pueden ser validadas directamente ya que las propiedades de la distribución de entrada no son conocidas *a priori*.
- Necesitan el conocimiento de un experto tanto en la técnica utilizada de visualización, como en el dominio al que pertenecen los datos.

Esta tesis incluye el análisis de una distribución de datos reales de habla discreta. El conjunto de datos consiste en una serie de vectores de características, obtenidos a partir de un proceso de extracción paramétrica (que es comentada a continuación) sobre un subconjunto de trazas de habla de la base de datos de comandos aislados del proyecto IVORY [IVO]. En concreto, 80 locuciones de la palabra inglesa “right”. Este subconjunto fue grabado por locutores (principalmente no nativos) de diferente edad, sexo y variantes de habla, en la cámara semianecoica del Laboratorio de Comunicación Oral *Robert Wayne Newcomb* de la Facultad de Informática (Universidad Politécnica de Madrid), en dos modalidades diferentes: normal y “ruido inducido”. Ambos modos están caracterizados por un nivel de ruido ambiental prácticamente despreciable, con la diferencia de que en el segundo, al locutor se le induce ruido por medio de auriculares, obligándolo a pronunciar el comando de manera forzada (con mayor volumen, imitando la situación en la que una persona aumenta el volumen de su voz para ser escuchado en ambientes ruidosos). La Tabla 4.1 muestra el número de locuciones de la palabra “right” según sexo y modo de grabación.

Dada la enorme cantidad de información y complejidad asociada a las muestras originales de trazas de habla, es necesario recurrir a métodos de extracción paramétrica, que proporcionan vectores de unas pocas características en función de un análisis de segmentos cortos de las trazas (de 10 a 20 milisegundos), donde se puede suponer que la señal es cuasi-estacionaria y presenta unas propiedades constantes. De esta manera, a partir de estos tramos cortos de la señal, una nueva

Vectores de parámetros	Hombres	Mujeres
Normal	1232	2226
Ruido inducido	1313	2170

Tabla 4.2: Distribución de vectores de parámetros *RASTA-PLP* en la base de datos de habla.

representación comprimida es generada en forma de vectores de características (también denominadas rasgos, coeficientes, parámetros, etc.), más fácil de manejar, donde la información contenida en la traza original es resumida.

Existen varios algoritmos de extracción paramétrica que han sido aplicados a datos de habla y su calidad debe ser medida en el contexto de una tarea particular [Día99a]. En esta tesis se ha optado por trabajar con una extracción paramétrica con la que es posible obtener buenos índices de reconocimiento de habla (véanse [Ma95, SF96, Día99a]) en condiciones adversas: el método *Relative Spectral-Perceptual Linear Prediction (RASTA-PLP)* [HMBK91, HM94]. La herramienta utilizada se encuentra disponible en [URLg]. La técnica es una mejora del método predicción lineal perceptiva [Her90] (que a su vez combina características del análisis clásico de predicción lineal (*linear predictive coding (LPC)*) y bancos de filtros), y realiza un filtrado especial de los diferentes canales de frecuencias del analizador PLP. El resultado es un método menos sensible a ligeras variaciones del espectro, de manera que se produce una atenuación de la señal con poco cambio espectral sobre el tiempo, mientras que se resaltan las porciones con cambios que pueden indicar la iniciación de nuevos segmentos fonéticos. Los parámetros *RASTA-PLP* son dependientes del contexto precedente, posibilitando que las transiciones del habla queden mejor reflejadas [Día99a]. Esto hace pensar que la técnica puede ser útil para etiquetar trazas de habla. Las Secciones 6.3.3.16 y 6.3.3.17 presentan ejemplos de esta aplicación práctica, en la que intervienen dos nuevos métodos de visualización de MAO para facilitar aún más el proceso.

La base de datos está compuesta de un total de 6941 vectores de ocho parámetros *RASTA-PLP*, donde el primero es una estimación de la energía de la señal. La Figura 4.15 muestra la proyección de este conjunto de datos según sus dos primeras componentes principales. Cada componente o parámetro representa 10 milisegundos de la traza original y sus valores han sido normalizados de manera que la varianza final de cada componente es igual a uno. La frecuencia de muestreo de las trazas originales es de 11025 KHz. En general, las trazas están compuestas de una primera zona de silencio que dura aproximadamente 75 milisegundos, la palabra, y un silencio final, generalmente de duración mayor que 75



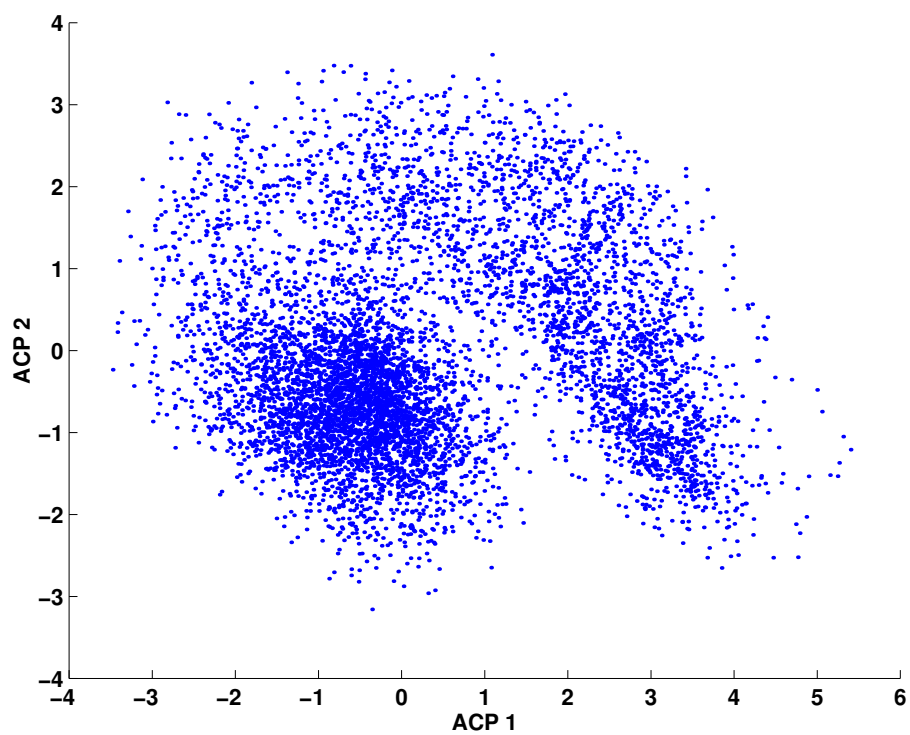


Figura 4.15: Proyección de la base de datos de la palabra “*right*” según sus dos primeras componentes principales.

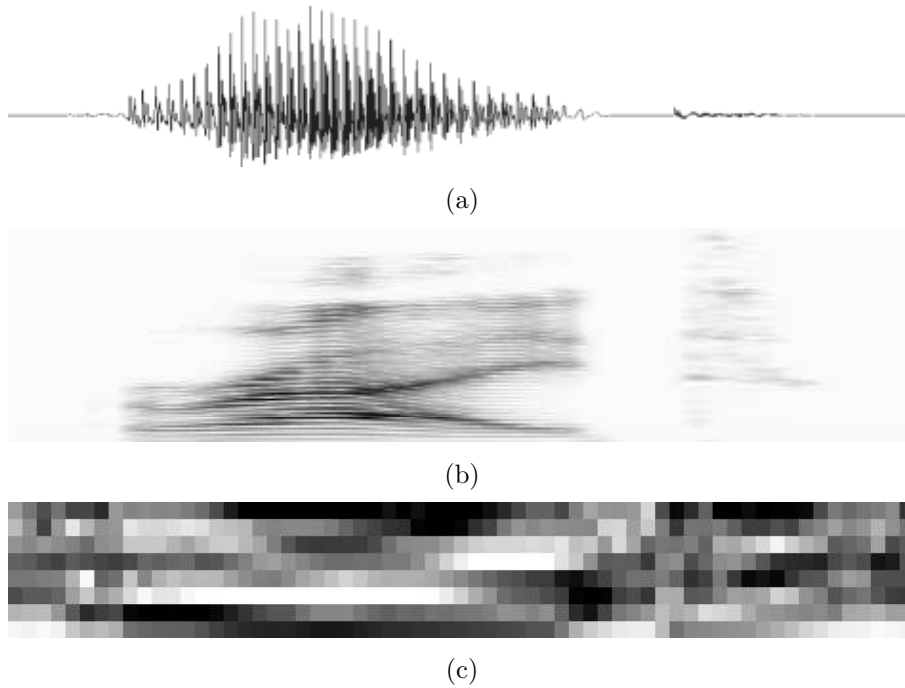


Figura 4.16: Traza de la palabra “right” (a), su espectrograma (b), y sus respectivos parámetros *RASTA-PLP* (c).

milisegundos. Esto tiene efectos sobre el análisis con MAO ya que una cantidad significativa de vectores de referencia se adaptan a este silencio. Por otro lado, debido a defectos en el proceso de segmentación de la base de datos original, algunas trazas pronunciadas por mujeres contienen una cantidad muy superior de silencio, contribuyendo aún más a estos efectos. La distribución de los vectores de parámetros según sexo y modo de grabación es mostrada en la Tabla 4.2.

La Figura 4.16 muestra una traza de la palabra “right” que corresponde a un locutor masculino grabado sin ruido inducido (a), su espectrograma (b), y sus respectivos parámetros *RASTA-PLP* en orden creciente de abajo a arriba, donde niveles de gris oscuros representan valores altos de los coeficientes (c).

Para ilustrar las características de algunos métodos de visualización de MAO en el Capítulo 6 se emplea un MAO  $(16 \times 16)$ -dimensional entrenado con esta base de datos. El MAO particular ha sido escogido entre varios, buscando un equilibrio entre un error de cuantización y un error topográfico (véase (3.16)) bajos.

## Capítulo 5

# MÉTODOS TRADICIONALES DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS

Los MAO encajan adecuadamente en un marco de AED ya que su propósito principal es la visualización y análisis de relaciones no-lineales de datos multidimensionales [Koh01]. A pesar de la importancia de la tarea de visualización, las técnicas gráficas para analizar MAO no son abundantes. Incluso el propio Kohonen sólo dedica un breve apartado a los métodos de visualización en su libro más reciente sobre MAO [Koh01]. Por el contrario, este capítulo lleva a cabo una extensa revisión y recopilación de las técnicas de visualización de MAO más relevantes de la literatura. En general, las técnicas son utilizadas para visualizar los patrones de datos, los vectores de referencia, la estructura de *clusters* y/o los componentes de los vectores de referencia.

Aunque es posible adaptar algunas técnicas para ser empleadas con tecnología de visión tridimensional, el estudio está restringido al análisis de MAO sobre soportes bidimensionales, como el papel o la pantalla de un monitor, en los que la visualización de MAO de dimensión superior a dos es problemática [Ves00]. De esta manera, los métodos son descritos principalmente para MAO de dimensión menor o igual a dos (que son los más utilizados por las razones mencionadas en la Sección 3.2.5.1). Cabe destacar que la capacidad de análisis con estos MAO es limitada, ya que un requisito general de los métodos de visualización es que

éstos se adapten adecuadamente a la distribución de datos. En otras palabras, la proyección debe preservar la topología, con lo cual la dimensión del MAO debe coincidir (lo mejor posible) con la dimensión intrínseca de los datos. De esta manera, los MAO bidimensionales pueden presentar deficiencias cuando la dimensión intrínseca de la distribución de datos es superior a dos.

Los tres primeros apartados de este capítulo describen técnicas tradicionales de visualización de información en MAO. La mayoría de éstas fueron propuestas en la primera mitad de la década de los noventa, en plena revolución informática asociada a interfaces gráficas de usuario. Son las más utilizadas en la práctica y destacan por su eficacia y sencillez, posiblemente condicionada por el estado de la tecnología, relativamente poco desarrollada por entonces, relacionada con la generación y visualización de imágenes por computador. De esta manera, suelen representar información mediante imágenes monocromáticas en las que aparece la estructura de la topología del MAO (por ejemplo, una cuadrícula cuando la topología es rectangular).

Los siguientes apartados abordan otras técnicas más recientes y sofisticadas, que incluyen estrategias híbridas para combinar varios métodos anteriores, métodos de proyección de datos y/o vectores de referencia, incorporación de color, combinación con otras técnicas de proyección y diversas mejoras.

Los nuevos métodos propuestos en esta tesis, que siguen dichas líneas recientes de investigación, son descritos en el Capítulo 6.

## 5.1. Visualización de Distancias

En general, la mayoría de métodos de visualización de MAO muestra información acerca de los *clusters* de la distribución de datos. Este apartado describe los métodos más populares para detectar la estructura de *clusters*, los cuales están basados en representar las distancias entre los vectores de referencia de las neuronas en el espacio de observación o mapa.

Cuando un MAO se adapta a una distribución de datos, los vectores de referencia tienden a agruparse donde más datos hay, es decir, en los *clusters*. Por esta razón los MAO pueden reflejar adecuadamente su estructura. La Figura 5.1 muestra la adaptación final de un MAO (20 × 20)-dimensional a la distribución 8-dimensional compuesta por ocho *clusters* disjuntos con forma hipercúbica. La

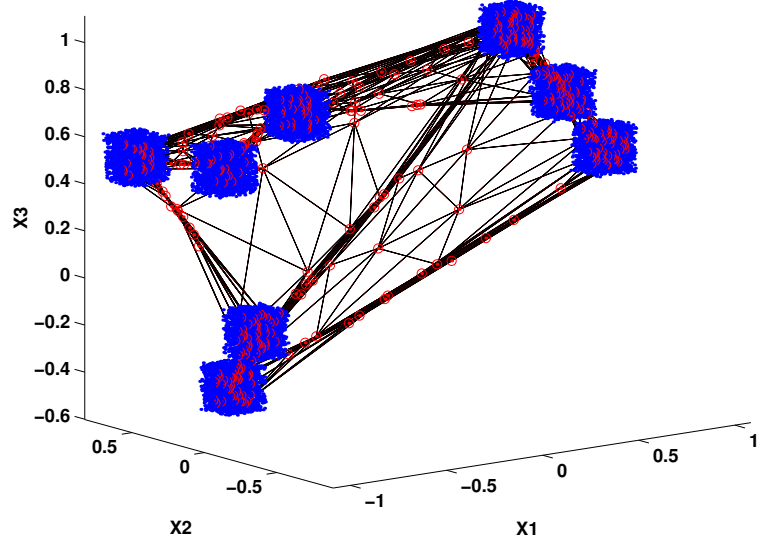


Figura 5.1: Adaptación de un MAO ( $20 \times 20$ )-dimensional a la distribución 8-dimensional (sólo las tres primeras componentes son visualizadas) formada por ocho *clusters* disjuntos con forma hipercúbica.

figura sólo muestra las tres primeras componentes del espacio de entrada. Queda claro que dentro de los *clusters* las distancias entre los vectores de referencia son pequeñas, mientras que fuera de ellos aparecen vectores de referencia que no representan a ningún dato o a muy pocos, encontrándose más alejados entre sí.

### 5.1.1. Rejilla de distancias

Una primera idea consiste en representar de alguna manera las distancias entre los vectores de referencia de neuronas vecinas en el mapa. Esto puede ser realizado de varias maneras, por ejemplo, representando dichas distancias mediante segmentos coloreados con un determinado grado de intensidad luminosa (véase la Figura 5.2). En esta tesis la estrategia general a la hora de transformar los valores de una determinada variable  $x$  a valores de intensidad luminosa (niveles de gris u otro color)  $I(x)$  consiste en aplicar la siguiente fórmula lineal:

$$I(x) = \frac{255}{x_{\text{máx}} - x_{\text{mín}}}(x - x_{\text{mín}}) \quad (5.1)$$

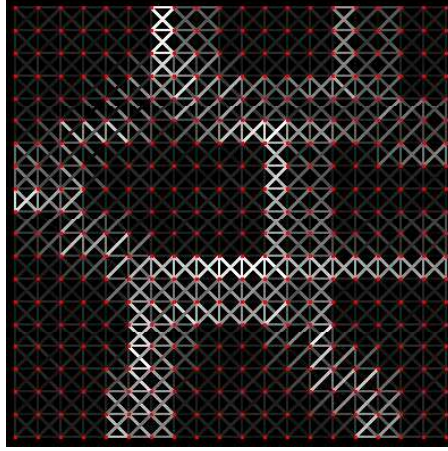


Figura 5.2: Rejilla de distancias para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica.

donde 255 es la máxima intensidad luminosa posible (codificada en un byte), y donde  $x_{\text{máx}}$  y  $x_{\text{mín}}$  representan los valores máximo y mínimo que toma la variable  $x$ , respectivamente. De esta manera, se aprovecha todo el rango dinámico de intensidad luminosa o brillo en la imagen. Sin embargo, los valores  $x_{\text{máx}}$  y  $x_{\text{mín}}$  pueden ser atípicos, en cuyo caso la imagen resultante presentaría poco contraste, dificultando su visualización e interpretación. En la práctica, es aconsejable tomar otros valores extremos para calcular las intensidades luminosas. Por otro lado, el resultado de este método y de varios descritos en este capítulo son imágenes monocromáticas, las cuales pueden ser mejoradas mediante técnicas convencionales de tratamiento de imágenes (suavizado, realce de bordes, operaciones sobre el histograma, etc. [Rus95, Cas96]).

En el ejemplo de la Figura 5.2 la asignación de colores es realizada según (5.1), de tal forma que los segmentos más claros corresponden a distancias relativamente elevadas y los oscuros a distancias relativamente pequeñas (la mínima distancia corresponde al negro y la máxima al blanco). En el ejemplo, las distancias “diagonales” son divididas previamente por el factor  $\sqrt{2}$ . Finalmente, zonas oscuras en el mapa corresponden a los posibles *clusters* mientras zonas claras representan bordes entre *clusters*. En la imagen es posible apreciar claramente que la distribución de entrada está formada por ocho *clusters* disjuntos.

Este tipo de representación, que denominamos “rejilla de distancias”, es muy similar a otras descritas en este apartado. Fue desarrollada cuando el autor de esta tesis abordó por primera vez el estudio de los MAO, sin conocimiento previo

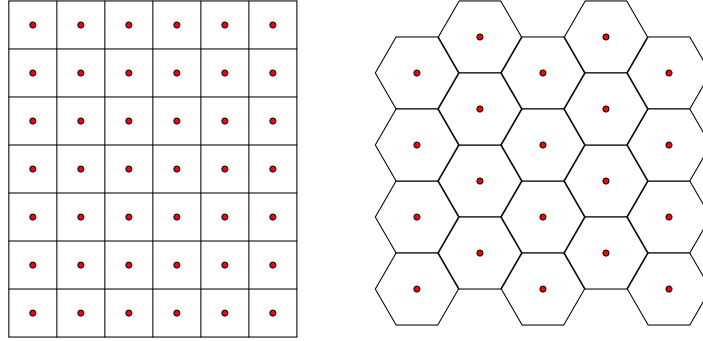


Figura 5.3: Configuraciones de las imágenes de las matrices de distancias.

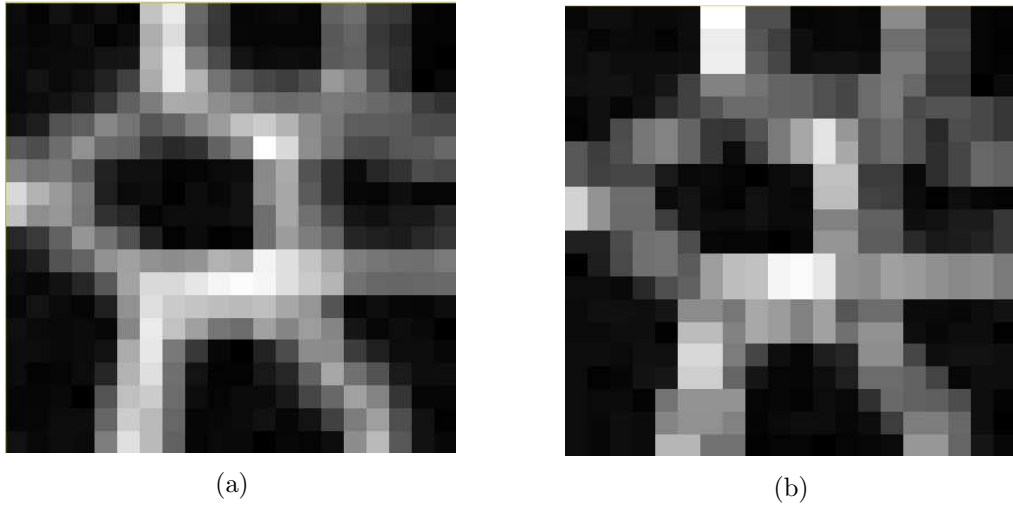


Figura 5.4: Matrices de distancias para un  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica. La imagen (a) está basada en la media de las distancias a todas las neuronas vecinas, mientras (b) lo hace en el valor máximo a las vecinas más próximas en las direcciones horizontal y vertical.

sobre las existentes técnicas de visualización. No aporta información significativa con respecto al resto de métodos presentados en este apartado, pero ayuda a entenderlos. Por otro lado, representa solamente las distancias entre los vectores de referencia de neuronas vecinas, y no resúmenes de éstas.

### 5.1.2. Matrices de distancias

Las imágenes de las matrices de distancias pueden ser vistas como simplificaciones de la rejilla de distancias y básicamente muestran la densidad de vectores de referencia en diferentes zonas del mapa. Primero el mapa es subdividido en una

serie de regiones asociadas a cada neurona, apareciendo un diagrama de Voronoi (regular) basado en las posiciones de las neuronas en el mapa. Por ejemplo, si la topología del mapa es cuadrada (rectangular) la imagen generada tendrá forma de cuadrícula (o matriz); si la topología es hexagonal la imagen se asemejará a un panal de abejas (véase la Figura 5.3). Posteriormente las celdas de Voronoi asociadas a cada neurona son coloreadas con niveles de gris. La asignación de colores suele ser lineal, como en la rejilla de distancias, y está basada en resúmenes de las distancias entre los vectores de referencia de neuronas vecinas. Por ejemplo, la media, la mediana o el valor máximo a un determinado conjunto de neuronas vecinas [Ult92, Ult93, KMJ95, HD95, KMJ95]. Para MAO bidimensionales con topología rectangular se crea una matriz  $D$  donde sus elementos quedan definidos, por ejemplo, mediante:

$$d_{i,j} = \frac{1}{\text{card}(V_{i,j})} \sum_{\eta_{k,l} \in V_{i,j}} \|w_{i,j} - w_{k,l}\| \quad (5.2)$$

o

$$d_{i,j} = \max_{\eta_{k,l} \in V_{i,j}} \|w_{i,j} - w_{k,l}\| \quad (5.3)$$

donde  $V_{i,j}$  representa un conjunto determinado de neuronas adyacentes a la neurona  $\eta_{i,j}$  (en dirección horizontal, vertical y opcionalmente ambas diagonales), y  $\text{card}$  denota cardinalidad. La Ecuación (5.2) está basada en la media mientras (5.3) lo hace en el valor máximo. Posteriormente es posible generar una imagen monocromática según el procedimiento descrito en (5.1), donde valores pequeños corresponden a tonos oscuros y elevados a claros.

La Figura 5.4 muestra una matriz de distancias basada en la media a todas sus neuronas vecinas (ocho en el interior del mapa, cinco en los bordes y tres en las esquinas) (a), y otra tomando el valor máximo de las distancias asociadas a las vecinas más próximas sin considerar las direcciones diagonales (cuatro en el interior del mapa, tres en los bordes y dos en las esquinas) (b). Los resultados son bastante similares, si bien la máxima distancia favorece el contraste de las imágenes [KMJ95]. El MAO corresponde al mismo utilizado en la Figura 5.2.

Aunque el uso del color es más habitual, los valores  $d_{i,j}$  asociados a cada neurona en las matrices de distancias pueden ser representados mediante otra característica. La Figura 5.5 muestra un ejemplo donde la matriz de distancias es visualizada mediante una rejilla regular hexagonal de celdas cuyo tamaño (perímetro o área) depende de los valores  $d_{i,j}$ .



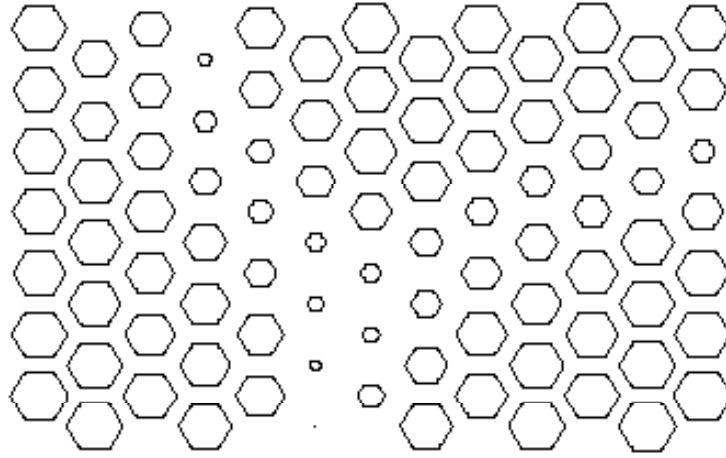


Figura 5.5: Matriz de distancias basada celdas de un determinado tamaño. Tomado de [Ves00].

Las imágenes de las matrices de distancias también pueden ser representadas mediante paisajes tridimensionales reflejando sus valores en un eje vertical [UV94, Ult99] (donde también es posible utilizar un código de colores como soporte a la visualización). La Figura 5.6 ilustra esta variante en un ejemplo clásico de visualización de datos mediante MAO. La distribución a analizar consiste en una cadena de dos anillos o eslabones (véase la Figura 4.5). La primera imagen de la Figura 5.6 muestra como un MAO se adapta a dicha distribución (para simplificar en la imagen el MAO es  $(20 \times 20)$ -dimensional) (a). En (b) la figura muestra la imagen bidimensional de una matriz de distancias basada en la media de las distancias a todas las vecinas de un MAO  $(64 \times 64)$ -dimensional adaptado a la distribución. Por último, la imagen (c) muestra la alternativa tridimensional. Los *clusters* o clases de la distribución pueden ser detectados si existen zonas asociadas a valles, las cuales aparecen rodeadas de colinas de una determinada altura. Existen varios algoritmos diseñados para calcular automáticamente los *clusters* de la distribución (véase la Sección 5.5.1). Por ejemplo, es posible detectar los dos eslabones de la distribución inundando el paisaje con un líquido hasta un cierto nivel [Ult99]. Nótese la dificultad que tienen otros métodos de clasificación para discriminar entre esas dos clases disjuntas. Las matrices de distancias han sido utilizadas extensamente en los últimos años en numerosas aplicaciones, donde se ha visto que aportan una visión muy eficaz de estructuras de dimensión elevada [Ult99].

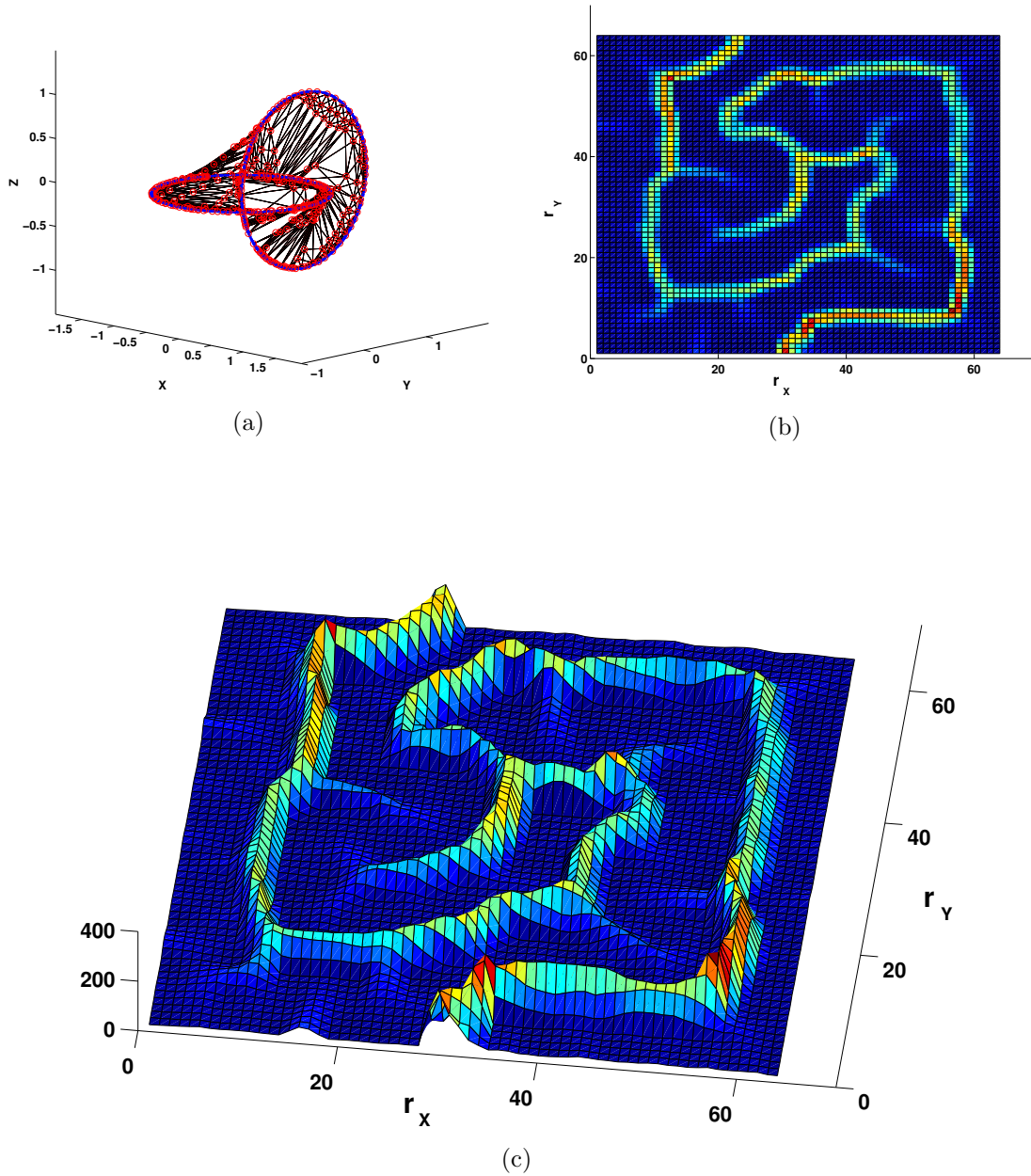


Figura 5.6: Adaptación de un MAO a la distribución formada por dos eslabones de una cadena (a). En (b) y (c) se muestran las matrices de distancias para un MAO ( $64 \times 64$ )-dimensional, en sus formas bidimensional y tridimensional, respectivamente.

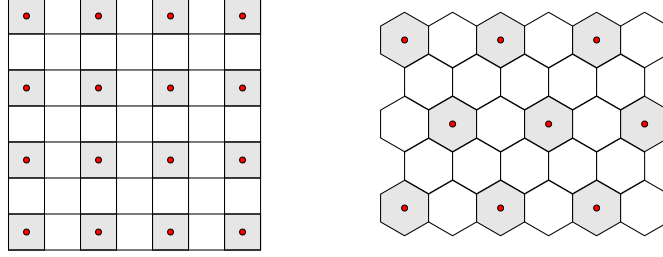


Figura 5.7: Configuraciones de las imágenes del método U-matrix.

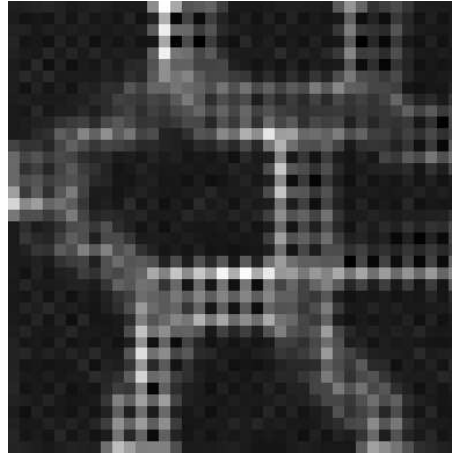
$u_{2x+1,2y}$	$=$	$\ w_{x,y} - w_{x+1,y}\ $	$x \in [0, X - 2] \quad y \in [0, Y - 1]$
$u_{2x,2y+1}$	$=$	$\ w_{x,y} - w_{x,y+1}\ $	$x \in [0, X - 1] \quad y \in [0, Y - 2]$
$u_{2x+1,2y+1}$	$=$	$\frac{\ w_{x,y} - w_{x+1,y+1}\  + \ w_{x,y+1} - w_{x+1,y}\ }{2\sqrt{2}}$	$x \in [0, X - 2] \quad y \in [0, Y - 2]$
$u_{2x,2y}$	$=$	Valor arbitrario	$x \in [0, X - 1] \quad y \in [0, Y - 1]$

Tabla 5.1: Definición de la matriz del método U-matrix.

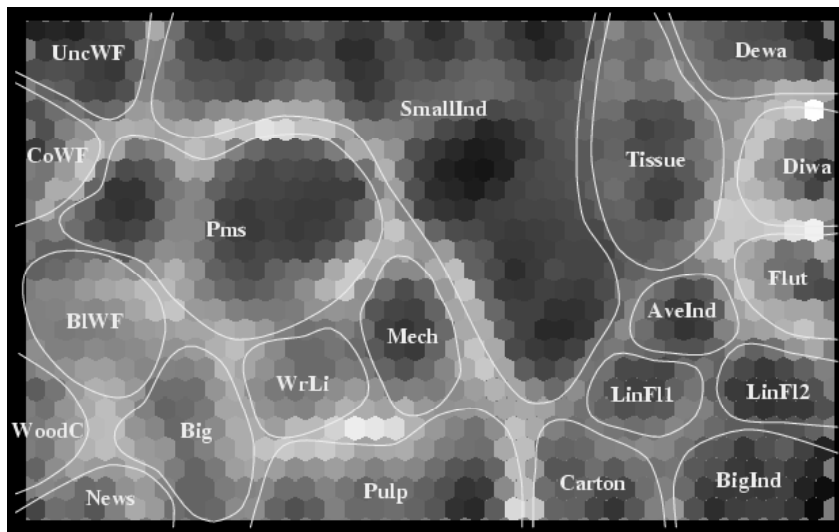
### 5.1.3. U-matrix

El primer y más popular método de análisis visual de MAO es la U-matrix o matriz de distancia unificada [US90], que es otro tipo de matriz de distancias. Representa prácticamente la misma información que la rejilla de distancias (cuando la topología es rectangular en vez de mostrar cada una de las distancias diagonales muestra una media ponderada de éstas), pero en el mismo formato que las matrices de distancias (cuadrícula o “panal de abejas”), véase la Figura 5.7. Crea imágenes de mayor resolución y aporta más información que las matrices de distancias presentadas en el apartado anterior al insertar celdas, con la forma de la topología del mapa, entre las celdas asociadas a las neuronas de manera que surge una nueva estructura regular.

Considérese un MAO con topología rectangular constituido por una rejilla de  $(X \times Y)$  neuronas. El método U-matrix crea una matriz  $U$   $((2X - 1) \times (2Y - 1))$ -dimensional cuyos elementos representan distancias entre vectores de referencia de neuronas vecinas. La definición de la matriz es descrita en la Tabla 5.1 [US90, RGD<sup>+</sup>02]. El valor de  $u_{2x,2y}$  es arbitrario (por ejemplo, es posible escoger la media, la mediana o la moda de los valores de celdas adyacentes, o incluso un valor constante). El factor  $\sqrt{2}$  en el denominador de  $u_{2x+1,2y+1}$  es opcional. Una definición de la U-matrix para MAO con topología hexagonal es descrita en [IKKK94].



(a)



(b)

Figura 5.8: Imágenes generadas mediante el método U-matrix. Análisis del MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos (a). U-matrix asociada a un MAO adaptado a datos de producción de una fábrica de papel y pulpa (b), tomado de [SVVH99].

La Figura 5.8 muestra dos ejemplos de imágenes generadas mediante este método. La primera (a) corresponde al mismo ejemplo utilizado para representar la rejilla de distancias y varias matrices de distancias (véanse las Figuras 5.1, 5.2 y 5.4). La segunda imagen (b) corresponde a un MAO entrenado con datos de producción de una fábrica de papel y pulpa, véase [SVVH99]. En estos casos las zonas oscuras también representan *clusters* y las claras bordes entre éstos. Para facilitar la interpretación del mapa los diversos *clusters* han sido etiquetados y resaltados.

### 5.1.4. Suma de todas las distancias

Existen otras alternativas a la hora de definir matrices de distancias. Por ejemplo, para cada neurona es posible tener en cuenta las distancias a vecinas únicamente según las conexiones (entre los vectores de referencia) del árbol generador mínimo. Este apartado propone una sencilla variante que consiste en representar mediante niveles de gris (u otra escala de colores), para cada neurona, la suma de las distancias entre su vector de referencia y el resto de vectores de referencia del MAO. Formalmente, considerando una MAO  $(X \times Y)$ -dimensional cuyos vectores de referencia son representados mediante  $w_{i,j}$ , se crea una matriz  $S$  de iguales dimensiones que el MAO donde cada elemento de la matriz queda definido mediante:

$$s_{i,j} = \sum_{k=1}^X \sum_{l=1}^Y \|w_{k,l} - w_{i,j}\| \quad (5.4)$$

Posteriormente es posible generar una imagen monocromática según el procedimiento descrito en (5.1). La imagen resultante requiere una interpretación diferente de los niveles de gris, en comparación con el resto de matrices de distancias descritas anteriormente. Esta imagen representa otra forma de analizar MAO donde las neuronas asociadas a un *cluster* suelen aparecer con un nivel de gris similar. De esta manera, los *clusters* pueden detectarse examinando las regiones donde el color presenta poca variación.

La Figura 5.9 muestra un ejemplo de este tipo de matriz de distancias, la cual es generada a partir del MAO  $(20 \times 20)$ -dimensional entrenado con la distribución de ocho *clusters* disjuntos. En la imagen es posible apreciar claramente los *clusters* y sus bordes, donde cada *cluster* o clase puede quedar definido mediante un determinado color o nivel de gris. Desafortunadamente, no es posible clasificar o etiquetar las neuronas en función de su color ya que diferentes *clusters* pueden

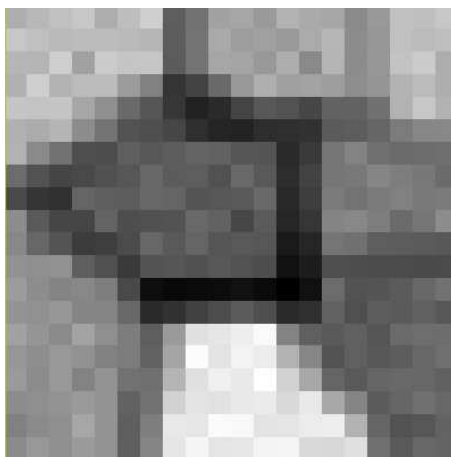


Figura 5.9: Representación de la suma de todas las distancias para el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos.

aparecer con el mismo color. La Sección 6.3 propone un nuevo método en el que todos los *clusters* suelen aparecer con colores diferentes, facilitando el etiquetado de las neuronas y el análisis trayectorias o secuencias de patrones de entrada.

### 5.1.5. Conexiones entre *clusters*

El método conexiones entre *clusters* [MR97] puede ser visto como una variante simplificada de la rejilla de distancias. Representa únicamente las conexiones horizontales y verticales entre neuronas vecinas mediante barras o segmentos coloreados, con un conjunto muy reducido de niveles de gris: negro para neuronas cuyos vectores de referencia son muy similares, un par de niveles de gris para distancias intermedias y blanco o simplemente ninguna conexión cuando los vectores de referencia se hallan muy separados. La Figura 5.10 muestra un ejemplo del método, donde la distribución contiene características de una serie de animales (véase [MR97]). En la imagen también aparecen algunas neuronas etiquetadas con los nombres de los animales asociados a ellas.

## 5.2. Visualización de Datos

Las imágenes generadas con los métodos anteriores son capaces de ilustrar estructuras esenciales de la distribución de datos. Sin embargo, pueden presentar deficiencias con determinadas distribuciones al no tener en cuenta los propios

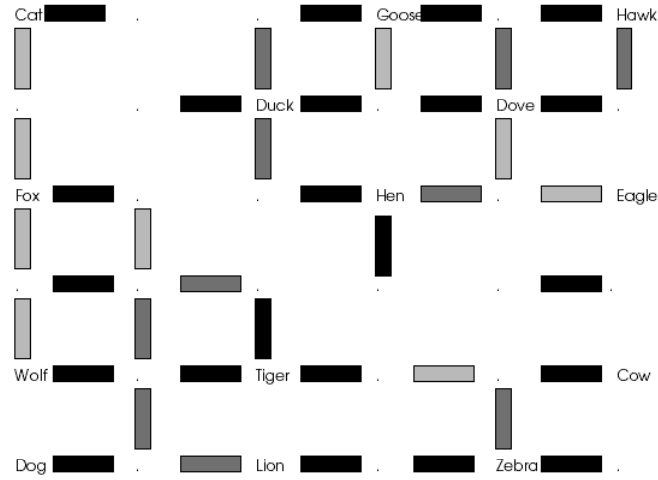


Figura 5.10: Conexiones entre *clusters*.

datos para crear las imágenes [RGD<sup>+</sup>02], además de necesitar generalmente un elevado número de neuronas para su correcta aplicación.

Para realizar un análisis más completo y robusto es necesario aportar información sobre las muestras de la distribución en las visualizaciones. De esta manera, es posible investigar la relación entre las diferentes regiones del mapa y los datos conocidos *a priori*, pudiendo comprobar si la proyección del MAO ha sido construida adecuadamente. Además de los datos de entrenamiento, la investigación de nuevos datos también es importante. Cuando las regiones del mapa son identificadas con clases conocidas se puede proceder a clasificar o a detectar nuevos tipos de datos [Ves00]. Este apartado repasa brevemente varias técnicas tradicionales para mostrar dicha información.

### 5.2.1. Unidad de máxima similitud y superficie de respuesta

Todas las técnicas de visualización de datos están basadas en la definición de la proyección de un patrón de entrada sobre el espacio de observación del MAO, y/o en su respuesta (salidas de las neuronas) ante un estímulo (patrón) de entrada. Generalmente, la posición de la unidad de máxima similitud (UMS) es utilizada para definir la proyección de un dato de entrada (véase la Sección 3.2.4.1). Sin embargo, no es habitual representar únicamente la UMS. Suele ser

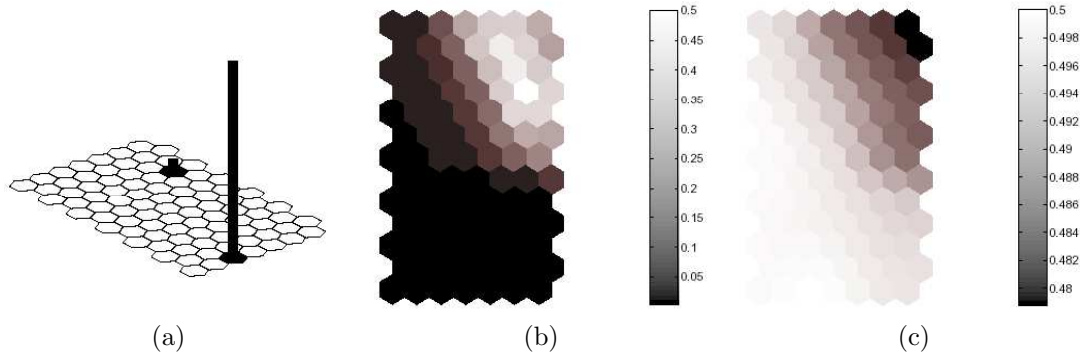


Figura 5.11: Análisis visual de dos unidades de máxima similitud (a), y respuesta difusa del MAO en ambos casos (b) y (c). Tomado de [Ves00].

útil medir también la precisión de la similitud de la UMS y la del resto de neuronas. La Figura 5.11 muestra diferentes formas de visualizar las unidades de máxima similitud. La imagen (a) visualiza la posición en el mapa de dos UMS para dos patrones de entrada, donde una barra vertical indica la precisión de la similitud. Su altura puede ser calculada teniendo en cuenta el cociente entre el error de cuantización  $q_i(x) = \|x - w_i\|$  de la unidad de mínima y máxima similitud:  $\min_i q_i(x) / \max_i q_i(x)$ . Las imágenes (b) y (c) son “superficies de respuesta” que muestran respuestas difusas de las neuronas a los dos patrones de entrada. Éstas han sido calculadas mediante  $1/(1 + (q_i(x)/a)^2)$ , donde  $a$  es una constante de normalización, véanse [Ves99, Ves00]. Las respuestas de las neuronas pueden ser calculadas también mediante un modelo generativo de mixturas [AHV99].

### 5.2.2. Histograma de datos

Otra técnica para detectar *clusters* y bordes entre éstos consiste en construir un histograma de datos. Se trata de una matriz  $H$  de iguales dimensiones que las del MAO. Los histogramas de datos visualizan el número o proporción de patrones de entrada asociados a cada neurona. Sea  $\Delta \subset \mathbb{R}^n$  el conjunto de entrenamiento del MAO (la distribución a analizar), cada elemento de la matriz del histograma de datos queda definido mediante:

$$h_{i,j} = \frac{1}{N} \text{card}\{x \in \Delta \mid \|x - w_{i,j}\| < \|x - w_{k,l}\| \quad \forall (k,l) \neq (i,j)\} \quad (5.5)$$

donde  $\text{card}$  denota cardinalidad y  $N$  es el número de datos de la distribución (que puede ser omitido).



$h'_{2x,2y}$	=	$h_{x,y}$	$x \in [0, X - 1]$	$y \in [0, Y - 1]$
$h'_{2x+1,2y}$	=	$\frac{h_{x,y} + h_{x+1,y}}{2}$	$x \in [0, X - 2]$	$y \in [0, Y - 1]$
$h'_{2x,2y+1}$	=	$\frac{h_{x,y} + h_{x,y+1}}{2}$	$x \in [0, X - 1]$	$y \in [0, Y - 2]$
$h'_{2x+1,2y+1}$	=	$\frac{h_{x,y} + h_{x,y+1} + h_{x+1,y} + h_{x+1,y+1}}{4}$	$x \in [0, X - 2]$	$y \in [0, Y - 2]$

Tabla 5.2: Suavizado especial de un histograma de datos.

$w'_{2x,2y}$	=	$w_{x,y}$	$x \in [0, X - 1]$	$y \in [0, Y - 1]$
$w'_{2x+1,2y}$	=	$\frac{w_{x,y} + w_{x+1,y}}{2}$	$x \in [0, X - 2]$	$y \in [0, Y - 1]$
$w'_{2x,2y+1}$	=	$\frac{w_{x,y} + w_{x,y+1}}{2}$	$x \in [0, X - 1]$	$y \in [0, Y - 2]$
$w'_{2x+1,2y+1}$	=	$\frac{w_{x,y} + w_{x,y+1} + w_{x+1,y} + w_{x+1,y+1}}{4}$	$x \in [0, X - 2]$	$y \in [0, Y - 2]$

Tabla 5.3: Inserción de vectores de referencia artificiales entre los originales.

La construcción y configuración de las imágenes es similar a las matrices de distancias (véanse las Figuras 5.3 y 5.6), y la información que aportan puede asemejarse bastante a la de los métodos basados en distancias, dado que los MAO tienden a representar la función de densidad de probabilidad de la distribución de datos de entrada. De esta manera, es posible generar imágenes monocromáticas que representen zonas de alta densidad de datos (posibles *clusters*) y zonas con pocos datos asociados (posibles bordes entre *clusters*).

La Sección 6.2 propone la fusión de histogramas de datos con matrices de distancias para obtener imágenes más completas y robustas. También pueden ser combinados con la U-matrix. En [RGD<sup>+</sup>02] se proponen varias formas de crear histogramas de datos con la misma resolución que la U-matrix. Una de ellas consiste en realizar un suavizado especial del histograma de datos original  $H$  para generar una matriz  $H'$  de dimensión  $(2X - 1, 2Y - 1)$  definida en la Tabla 5.2, suponiendo que el MAO tiene topología rectangular con  $X$  filas e  $Y$  columnas. Una segunda alternativa útil cuando el MAO es pequeño consiste en generar un nuevo MAO insertando neuronas artificiales entre las originales y calculando los nuevos vectores de referencia  $w'$  según la Tabla 5.3, para posteriormente generar un histograma de datos clásico en función de este nuevo MAO.

La Figura 5.12 muestra un histograma de datos clásico con la resolución de las matrices de distancias (a) y otro suavizado de mayor resolución (b), para el MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos. Los ocho *clusters* aparecen nítidamente, al igual que los bordes, que

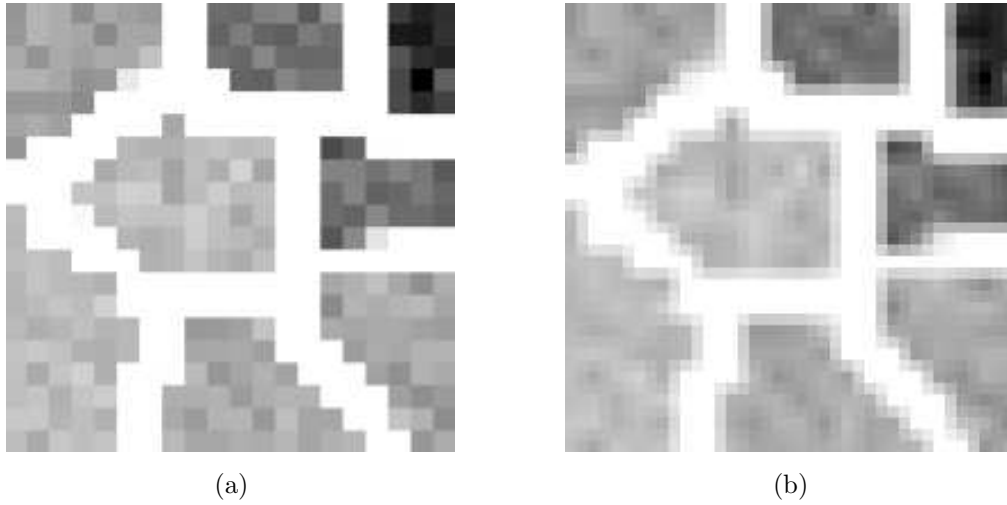


Figura 5.12: Histogramas de datos para el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos. Resolución normal donde cada neurona queda representada por una celda (a), e imagen suavizada con la misma resolución que la imagen del método U-matrix (b).

aparecen en blanco. También es posible apreciar como los *clusters* en la parte superior-derecha, que son más pequeños, aparecen más oscuros. Esto es debido a que los ocho *clusters* contienen la misma cantidad de datos, por lo que las neuronas de dichos *clusters* representan a más patrones de entrada.

Generalmente, dada la estructura ordenada de los vectores de referencia, cuando un MAO es entrenado aparecen neuronas interpolantes que corresponden a nodos de la red cuyos vectores de referencia tienen una proporción muy pequeña o nula de datos asociados. Estas neuronas pueden ser marcadas de forma especial en los histogramas de datos o en otras visualizaciones.

### 5.2.3. Interpolación básica

A diferencia de los métodos de visualización de datos que proyectan los patrones sobre la rejilla discreta del MAO, es posible variar la definición de la proyección de un dato sobre el espacio de observación de modo que éste último sea continuo. El problema puede ser enfocado desde el punto de vista de la interpolación. En otras palabras, se puede aprovechar la correspondencia discreta  $f_d$  entre el conjunto  $\Omega$  de vectores de referencia del MAO y el conjunto  $\Phi$  de las posiciones de las neuronas (véase (3.1)), para definir una función, correspondencia o proyección  $f$  del espacio continuo de los datos  $\mathbb{R}^n$  al espacio continuo de

observación  $\mathbb{R}^m$ :

$$f : \mathbb{R}^n \rightarrow \mathbb{R}^m \quad (5.6)$$

Esta estrategia incrementa la resolución de los histogramas de datos y puede ayudar a encontrar y descubrir nuevas e interesantes estructuras en los datos. Encaja bien en el marco del AED ya que muestra todos los datos de la distribución, evitando la pérdida de información debida a la cuantización.

La forma más sencilla de obtener una visión de todos los datos consiste en proyectarlos sobre la posición de la unidad de máxima similitud y añadir un pequeño desfase o error aleatorio [Ves00]. Formalmente la proyección puede ser descrita mediante:

$$f(x) = r_c + \varepsilon_{x,c} \quad (5.7)$$

donde  $r_c$  es la posición en el mapa de la UMS correspondiente al patrón de entrada  $x$ , y  $\varepsilon_{x,c}$  es un vector de desfase. En principio, tanto el módulo como el ángulo de este desfase pueden ser escogidos de manera aleatoria, restringiendo la ubicación del vector proyectado a la celda de Voronoi de la neurona ganadora en el espacio de observación. De esta manera, la imagen resultante es similar al histograma de datos ya que ilustra la concentración de datos en torno a cada neurona.

Para no perder información sobre la precisión de las similitudes, opcionalmente el módulo del desfase puede ser proporcional a la distancia entre el patrón de entrada y el vector de referencia de la UMS. Por ejemplo, si el dato coincide con un vector de referencia, el módulo del desfase debe ser igual a cero, con lo que sería proyectado sobre la posición de la neurona ganadora. De esta manera, se obtiene una imagen de la dispersión de datos relativa cerca de los vectores de referencia, es decir, una medida visual del error de cuantización asociado a cada neurona. Obsérvese que es posible generar una imagen equivalente monocromática según la filosofía de las matrices de distancias o los histogramas de datos, donde el nivel de gris de cada celda corresponda al error de cuantización relativo a cada neurona. La Figura 5.13 muestra un ejemplo de esta sencilla forma de proyectar los datos sobre el mapa. La distribución de datos corresponde a la formada por una circunferencia, un segmento, un cuadrado y un ortoedro, a la que se ha adaptado un MAO (20 × 20)-dimensional (a). La matriz de distancias (b) y el histograma de datos (c) indican que la distribución se ha adaptado según la diagonal secundaria (45°), donde la circunferencia aparece en la esquina superior-derecha y el ortoedro en la inferior-izquierda. El histograma de datos indica que las neuronas situadas

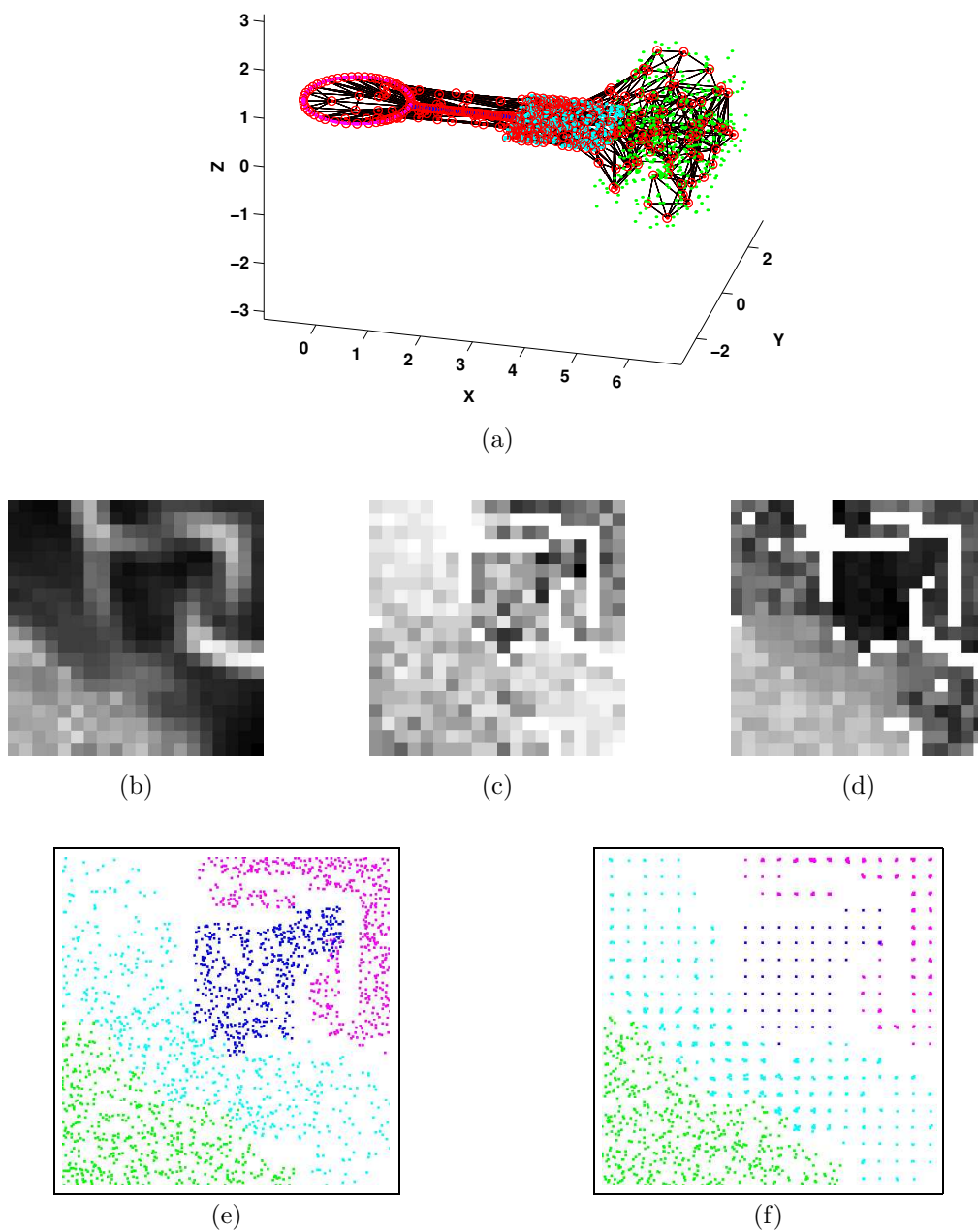


Figura 5.13: Interpolación básica. MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por cuatro clases de diferente dimensión intrínseca (a). Matriz de distancias (b). Histograma de datos (c). Visualización del error de cuantización de cada neurona con niveles de gris (d). Proyección aleatoria (e). Proyección basada en la distancia desde cada dato a su vector de referencia más cercano (f).

en las otras dos esquinas apenas representan datos, a pesar de que sus vectores de referencia se hallan próximos, como indica la matriz de distancias. En (d) la imagen monocromática indica el error de cuantización relativo de cada neurona, donde niveles oscuros y claros representan errores de cuantización bajos y altos, respectivamente; y donde el blanco indica ausencia de datos. La interpolación aleatoria (e) prácticamente aporta la misma información que el histograma de datos. Por otro lado, la interpolación basada en el error de cuantización (f) es equivalente a (d), e indica los diferentes grados relativos de dispersión de datos en cada zona del mapa. En otras palabras, ilustra la calidad de la adaptación de los vectores de referencia a los datos. En este ejemplo el error de cuantización es mayor según aumenta la dimensión intrínseca de los datos. Aunque la interpolación básica no considera la coloración de los datos proyectados las dos últimas imágenes han sido coloreadas para facilitar la ubicación de las diferentes clases en el mapa.

Estas estrategias simples de interpolación presentan varios problemas. En primer lugar, sólo tienen en cuenta la posición de la UMS, por lo que no pueden determinar un ángulo apropiado del desfase. Este problema se acentúa si el MAO es pequeño, al requerir mayor precisión y calidad en la proyección. Por otro lado, la proyección no es una función, ya que los algoritmos pueden proyectar un mismo vector de entrada a varias posiciones. Por otro lado, la proyección es discontinua en el sentido de que dos patrones muy próximos en el espacio de los datos pueden ser proyectados a posiciones muy diferentes en el mapa (por su naturaleza aleatoria y porque dos datos próximos pueden ser representados por dos vectores de referencia diferentes). Su utilidad también es limitada ya que existen métodos equivalentes que aportan la misma información.

La Sección 5.4.4 describe un método de interpolación basado en coordenadas afines de sistemas locales [GR93, GR95a, GR95b], el cual considera tres neuronas para calcular el ángulo y módulo del desfase. Sin embargo, este método también presenta diversos problemas que limitan su aplicación para ciertas distribuciones. La Sección 6.1 propone un nuevo método de interpolación [RDG<sup>+</sup>02, RG03] que puede tener en cuenta todas las neuronas para calcular la proyección, resolviendo la mayoría de problemas asociados a estos últimos métodos.

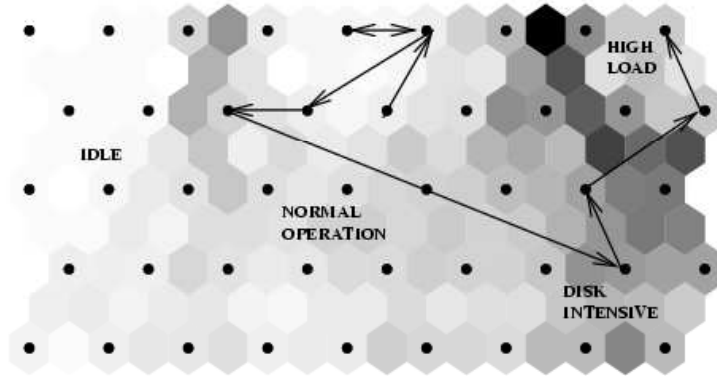


Figura 5.14: Visualización de trayectorias. Tomado de [SVVH99].

#### 5.2.4. Visualización de trayectorias

Cuando el problema consiste en analizar una secuencia de patrones de entrada una solución consiste en calcular las neuronas ganadoras  $w_c(t)$  y conectarlas mediante un grafo dirigido que indique una trayectoria en el mapa, véase [KKS92]. De esta manera, es posible estudiar el comportamiento de un proceso en el tiempo. La Figura 5.14 ilustra esta idea, en la que diversas zonas del mapa aparecen etiquetadas para facilitar la determinación del estado de un proceso. Los datos utilizados en este ejemplo pertenecen a varias medidas asociadas a un computador en un entorno de red, como tasas de utilización de la unidad central de proceso y el volumen de tráfico en la red.

La identificación de trayectorias o determinadas secuencias de patrones de entrada puede estar basada en imágenes de mapas en las que las neuronas de los *clusters* aparecen con un mismo color [RGG01, RG03], como las generadas por los métodos de contracción (véanse las Secciones 5.5.2.3 y 6.3). De esta manera, las trayectorias pueden ser analizadas mediante secuencias de iconos coloreados, donde el color de cada icono corresponde al de la neurona ganadora del dato de la secuencia asociado a dicho icono.

#### 5.2.5. Etiquetado

Otro método tradicional de observación basado en los propios datos hace uso de etiquetas asociadas a varias neuronas del mapa. Después de entrenar un MAO los datos conocidos *a priori* de varias clases son introducidos para detectar las posiciones de sus neuronas ganadoras. De esta manera, es posible etiquetar dichas

neuronas con nombres de clases conocidas y visualizar la distribución de éstas a lo largo del mapa. El proceso de etiquetado generalmente es manual (al ser supervisado) aunque existen trabajos que lo automatizan, por ejemplo el método “labelSOM” [Rau99, RM99], que ilustra los nombres de las variables de los vectores de entrada que más se asemejan a los del vector de referencia que representa a dichos datos de entrada.

La Figura 5.15 muestra dos ejemplos de mapas etiquetados. El “mapa fonético” mostrado (a), también denominado “maquina de escribir fonética” [Koh88], es el resultado de entrenar un MAO con un conjunto de vectores de coeficientes cepstrales tomados de trazas de habla continua finlandesa de un hablante. Las neuronas son etiquetadas con el fonema que ocurre más frecuentemente cuando éstas son las ganadoras. El “mapa de pobreza” (b), véase [KK96], describe la calidad de vida en una serie de países, cuyos nombres aparecen abreviados sobre ciertas neuronas, mientras que las neuronas que no representan a ningún país se muestran mediante puntos. En ambos casos la topología del mapa es hexagonal y las imágenes han sido suavizadas de manera que los niveles de gris representan el grado de agrupación o *clustering*. Al igual que en el resto de ejemplos, las zonas oscuras representan *clusters* y las claras bordes entre éstos. El suavizado es una simple operación gráfica que genera una variante que puede facilitar la interpretación del mapa.

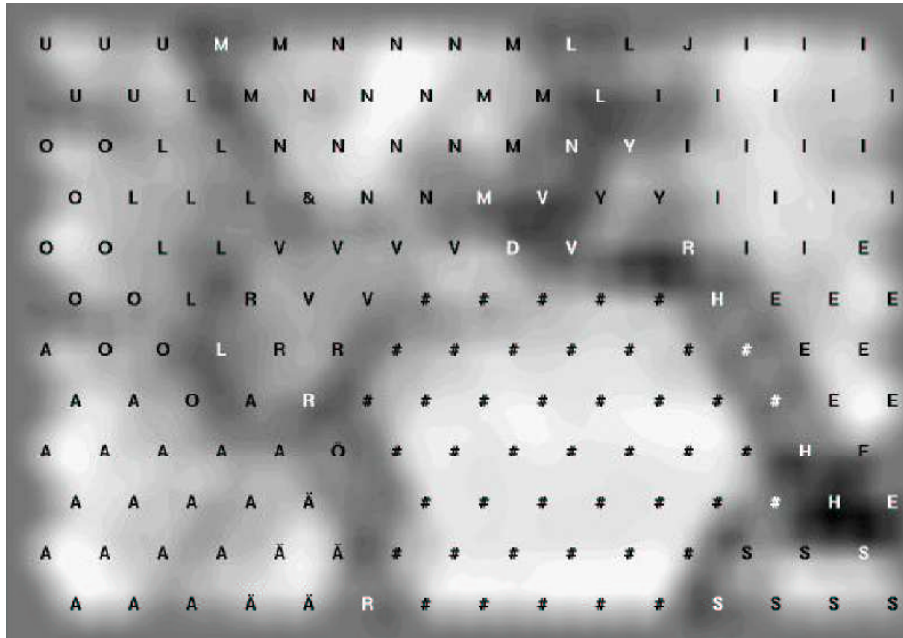
## 5.3. Visualización de Componentes

Una vez analizada la estructura general de la distribución en el mapa es posible centrarse en detalles más pequeños [Ves00], por ejemplo:

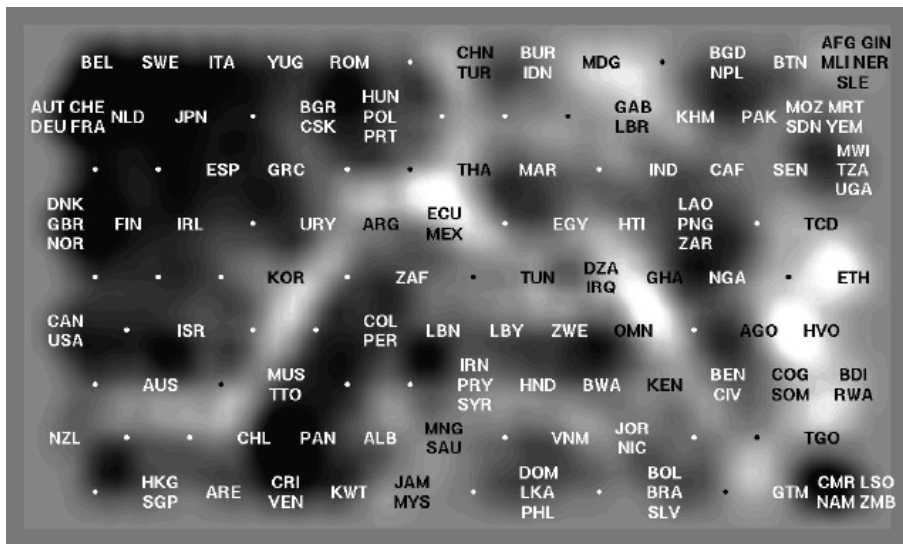
- ¿Qué valores toman las diferentes variables?
- ¿Qué combinaciones de valores aparecen en diferentes partes del mapa?
- ¿Existen dependencias significativas entre las variables?

Para dar respuesta a las cuestiones anteriores los métodos tradicionales están basados en la visualización de los valores de los componentes de los vectores de referencia del MAO.

## 5. MÉTODOS TRADICIONALES DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS



(a)



(b)

Figura 5.15: Etiquetado de MAO sobre imágenes (suavizadas) generadas mediante el método U-matrix. “Mapa fonético” (a) y “mapa de pobreza” (b). Tomados de [KVK00].



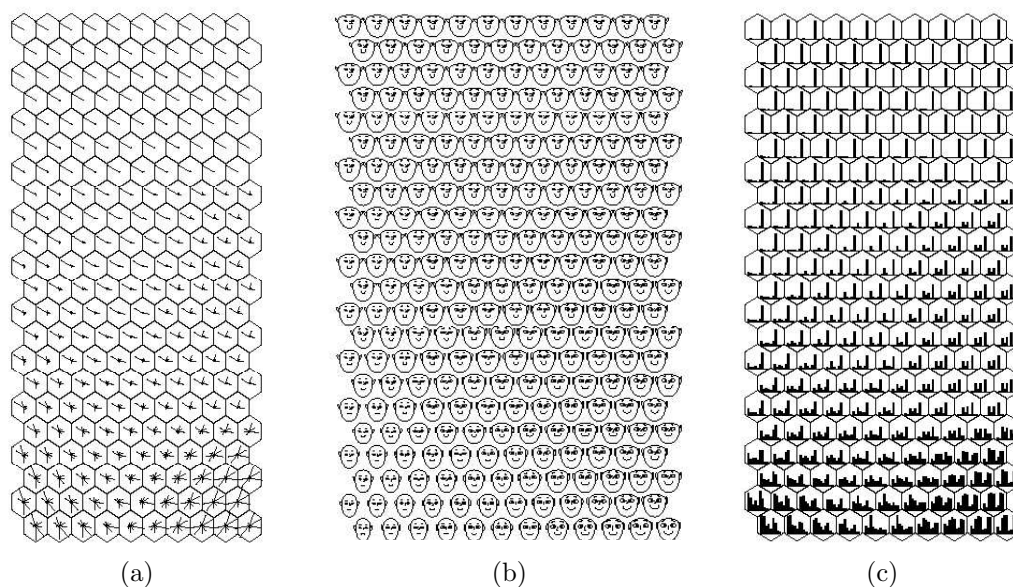


Figura 5.16: Visualización directa de vectores de referencia: iconos de estrellas (a), caras de Chernoff (b), e iconos de columnas (c). Tomado de [Ves02].

### 5.3.1. Visualización directa

Aprovechando la estructura regular de la rejilla del MAO y la ordenación de los vectores de referencia, éstos pueden ser visualizados y comparados directamente en el mapa mediante las técnicas de visualización de datos de dimensión elevada descritas en la Sección 2.5.2.2, véanse [HK97, Ves02]. La Figura 5.16 muestra tres tipos de visualización directa vectores de referencia: iconos de estrellas (a), caras de Chernoff (b), e iconos de columnas (c).

### 5.3.2. Planos de componentes

La técnica más popular para visualizar las variables de los vectores de referencia de los MAO son los planos de componentes. Para cada variable generan una rejilla o matriz donde ilustran su valor para todas las neuronas del MAO (por ejemplo, mediante un código de colores como una escala de grises). Es posible apreciar relaciones (dependencias o correlaciones) entre variables cuando aparecen estructuras similares en idénticos lugares de los planos de componentes. En estos casos, cuando cambia el valor de una variable la otra debe cambiar también. Sin embargo, la sensación visual de dependencia puede dar falsas pistas, por lo que es necesario validar las dependencias encontradas mediante otros métodos

[LK00]. Los planos de componentes suelen ser representados simultáneamente. Junto con una técnica de visualización de *clusters* es posible observar el valor que toma cada variable en cada *cluster*. Por otro lado, éstos pueden ser utilizados para visualizar MAO tridimensionales, mostrando un plano de componente por cada capa horizontal de neuronas [Kiv98].

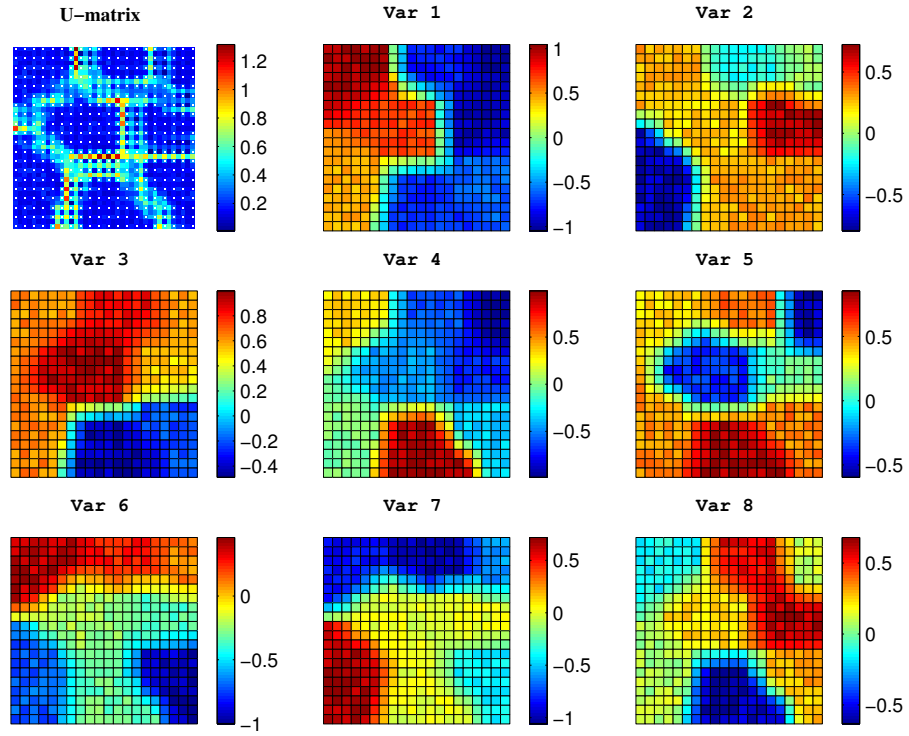
La Figura 5.17 muestra la U-matrix y los planos de componentes para dos MAO  $(20 \times 20)$ -dimensionales adaptados a la distribución 8-dimensional formada por ocho *clusters* disjuntos (a), y a la distribución tridimensional formada por dos eslabones encadenados (b). En estos ejemplos no se observan dependencias entre las variables, aunque en (a) es posible intuir la configuración de los *clusters* en el mapa (nótese que en este caso concreto, los *clusters* podrían ser visualizados mediante la superposición de algunos planos de componentes), y en (b) las zonas de color constante (verde, que corresponde al valor cero) en la primera y tercera variable indican los dos eslabones (clases) existentes, ya que éstos están definidos en el plano  $YZ$  ( $X = 0$ ) y  $XY$  ( $Z = 0$ ).

Por otro lado, los planos de componentes pueden ser combinados con otros métodos como matrices de distancias [HK97] (véase la Figura 5.18) o histogramas de datos [SVVH99] (véase la Figura 5.19), para incluir más información en una misma imagen.

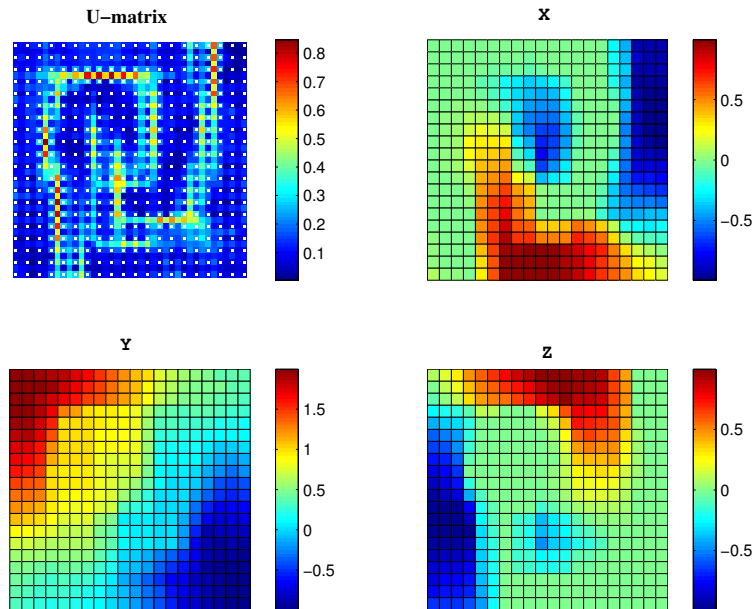
Finalmente, si los *clusters* de la distribución son disjuntos otra opción consiste en representar la media de los planos de componentes. La imagen final puede representar cada *cluster* con un nivel de gris diferente. La Figura 5.20 muestra un ejemplo de este método con un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos.

## 5.4. Métodos Refinados

Aproximadamente a partir de 1997 se empezó a desarrollar nuevas técnicas mejoradas de visualización de MAO, algo más sofisticadas que las presentadas en los apartados anteriores. Este punto de inflexión puede ser debido al desarrollo de nueva tecnología en cuanto a interfaces gráficas, librerías gráficas, lenguajes y entornos de programación visuales que facilitan y motivan en gran medida el uso de técnicas gráficas; lo que también justificaría la escasez de métodos hasta entonces. Por ejemplo, el uso de colores en gráficas estadísticas fue abandonado durante mucho tiempo debido a la carencia de software y hardware para producir



(a)



(b)

Figura 5.17: U-matrix y planos de componentes para dos MAO ( $20 \times 20$ )-dimensionales adaptados a la distribución 8-dimensional formada por ocho *clusters* disjuntos (a), y a la distribución tridimensional formada por dos eslabones encadenados (b).

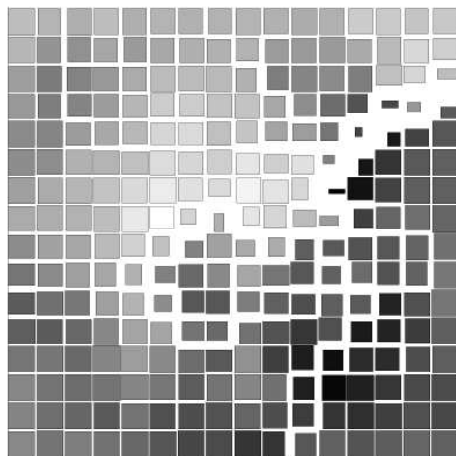


Figura 5.18: Plano de componente (niveles de gris) + matriz de distancias (representada por la distancia entre los cuadros, que indica la disimilitud entre los vectores de referencia de las neuronas). Tomado de [HK97].

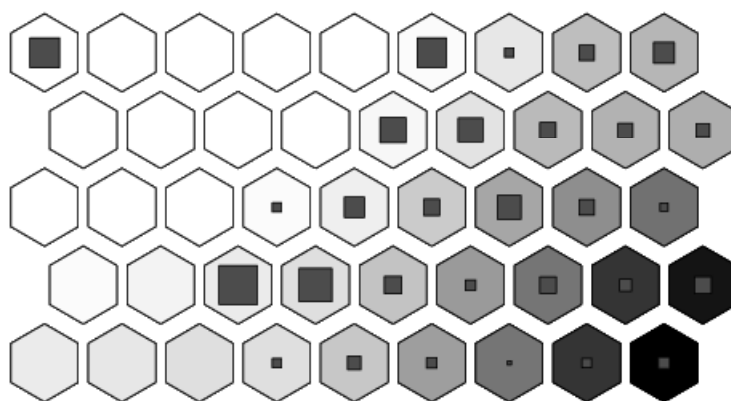


Figura 5.19: Plano de componente (niveles de gris) + histograma de datos (tamaño de los cuadros). Tomado de [SVVH99].

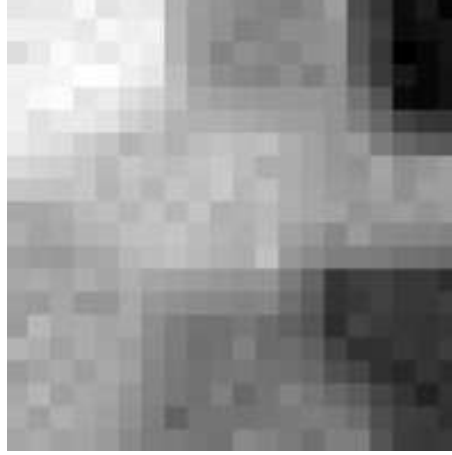


Figura 5.20: Media de los ocho planos de componentes relativos a un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos.

colores [YB95].

Los métodos presentados en este apartado introducen diversas mejoras en un intento por mitigar algunas de las limitaciones de los métodos tradicionales. Las nuevas técnicas están basadas en definiciones más sofisticadas de la proyección de una patrón sobre el mapa; o tratan de combinar información de distancias, de los propios datos y de sus variables. En general están orientados a visualizar la estructura de los *clusters* de las distribuciones.

### 5.4.1. Histogramas de datos suaves

Uno de los problemas relacionados con los histogramas de datos ordinarios es que no tienen en cuenta el hecho de que varias neuronas pueden representar adecuadamente un mismo vector de entrada. Los histogramas de datos suaves [PRM02] tratan de solucionar este problema. Su objetivo es visualizar los *clusters* de una distribución de dimensión elevada a través de la estimación de su función de densidad de probabilidad. Se trata de una estrategia difusa para ilustrar los *clusters* con niveles de gris de manera similar al método “generative topographic mapping” [BSW98]. El grado de pertenencia de un dato a cada neurona es controlado mediante un parámetro de suavizado  $s$ , el cual está basado en el orden de clasificación de las distancias entre el dato y los vectores de referencia de las neuronas. En concreto, el grado de pertenencia es  $s/c_s$ , para la neurona cuyo vector de referencia es el más próximo al dato;  $(s - 1)/c_s$ , para la segunda;  $(s - 2)/c_s$ ,

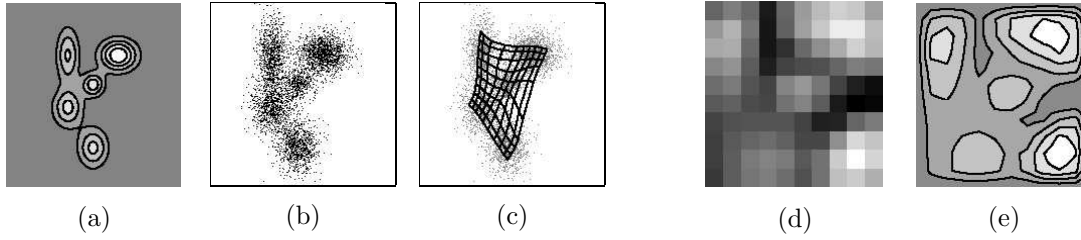


Figura 5.21: Histograma de datos suave. Cinco funciones de densidad de probabilidad Gaussianas (a), son estimadas a partir de los datos de una mixtura de cinco funciones Gaussianas (b), a los que se ha adaptado un MAO (c). El histograma de datos suave es mostrado en (d) y (e) mediante niveles de gris y un gráfico de contornos, respectivamente. Tomado de [PRM02].

para la tercera; etc. A partir de la  $s$ -ésima neurona, el grado de pertenencia es cero. La constante  $c_s = \sum_{i=0}^{s-1} s - i$ , asegura que la suma de pertenencias sea igual a uno. La página electrónica de [PRM02] contiene librerías para Matlab de los histogramas de datos suaves donde están implementadas otras estrategias para calcular el grado de pertenencia, por ejemplo, mediante el cálculo de los K-vecinos más cercanos.

La Figura 5.21 muestra un ejemplo del método aplicado a una distribución formada por una mixtura de cinco funciones Gaussianas. La figura ilustra las funciones de densidad de probabilidad estimadas (a), a partir de los datos de la distribución (b), que son utilizados para entrenar un MAO  $(10 \times 10)$ -dimensional (c). El histograma de datos suave para  $s = 8$  puede ser visualizado utilizando niveles de gris (d), o mediante una gráfico de contornos (e). En ambos casos las zonas claras corresponden a valores elevados del histograma de datos suave. Para  $s = 1$  el método corresponde al histograma de datos clásico. Véase [PRM02] para una descripción del ejemplo con varios valores de  $s$ .

#### 5.4.2. Contribución de un componente a la estructura de *clusters*

Un método computacionalmente muy simple que describe el poder discriminante de las variables de entrada al realizar la proyección del MAO es descrito en [KNK98]. Se trata de un método híbrido entre los planos de componentes y las matrices de distancias. Sirve para medir y visualizar la contribución explícita de los componentes originales a la estructura de *clusters* del MAO. El método genera imágenes monocromáticas de forma similar a la U-matrix, pero solamente

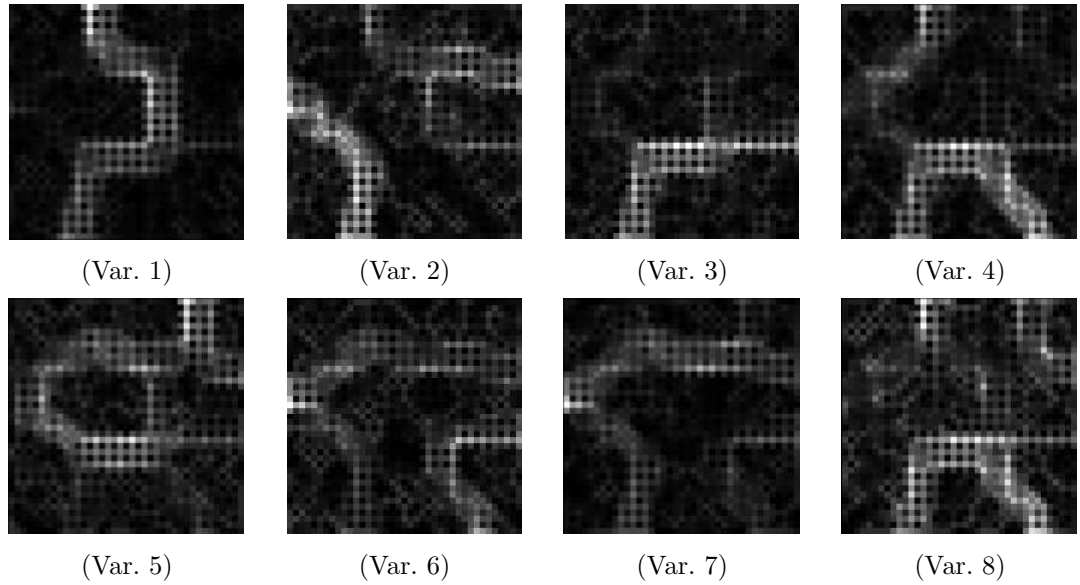


Figura 5.22: Contribución de las variables a la estructura de los *clusters* para el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos.

considera un componente. El método crea una “matriz de diferencias” teniendo en cuenta la diferencia entre un solo componente de los vectores de referencia de neuronas vecinas, en vez de calcular diferencias vectoriales con todos los componentes. Posteriormente, es posible calcular la correlación entre la U-matrix y las matrices de diferencias. Si ésta es elevada para una determinada variable (componente), en una determinada zona del mapa, su contribución a la estructura de *clusters* será significativa y tendrá un elevado poder explicativo en ese dominio de valores. El poder discriminante de las variables queda manifiesto especialmente en los bordes entre los *clusters*, donde suelen aparecer las mayores diferencias entre *clusters* vecinos [Koh01].

La Figura 5.22 muestra la contribución de todas las variables a la estructura de *clusters* del MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos, donde los bordes entre los *clusters* quedan claramente resaltados.

### 5.4.3. Análisis de bordes

Las imágenes generadas por las matrices de distancias pueden ser ruidosas si no son tratadas con información adicional. Una estrategia para mejorarlas consiste en combinarlas con información acerca de los propios datos de la distribución.

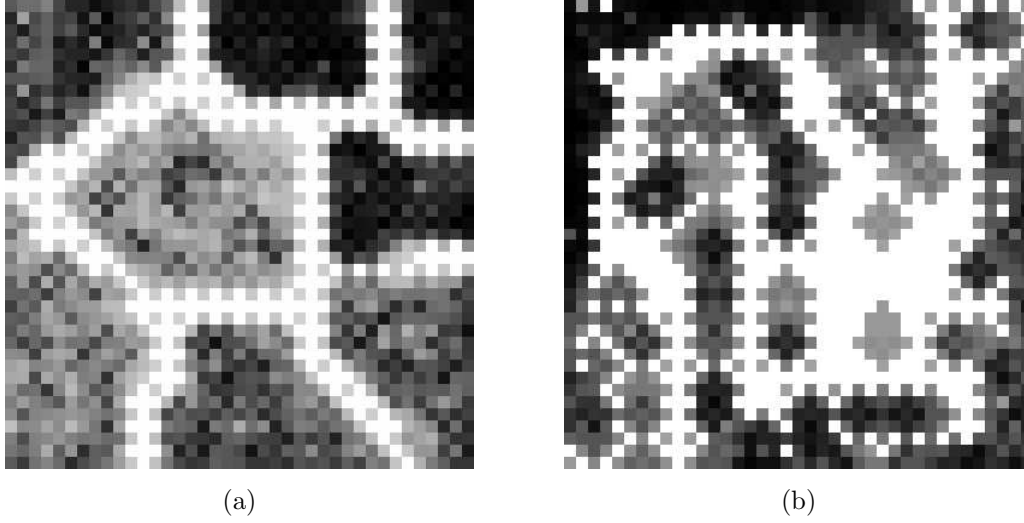


Figura 5.23: Análisis de bordes para los MAO (20×20)-dimensionales adaptados a la distribución formada por ocho *clusters* disjuntos (a), y a la distribución formada por dos eslabones de una cadena (b).

En esta línea, un método que facilita el análisis de bordes entre los *clusters* utilizando información sobre los datos en las vecindades de los vectores de referencia es propuesto en [KNK00, NTK<sup>+</sup>01]. El método busca dominios tanto de baja densidad de datos, como en los cuales el gradiente de la densidad de muestras cambia de dirección. Para cada vector de referencia  $w_i$ , calcula:

$$g_i = c_i - w_i \quad c_i = \frac{1}{N_i} \sum_{x_k \in V_i} x_k \quad (5.8)$$

donde  $c_i$  es el centroide de los  $N_i$  patrones cuyo vector de referencia más cercano es  $w_i$ , y  $V_i$  es la región de Voronoi asociada a  $w_i$ . El vector  $g_i$  apunta a la dirección donde la densidad de datos aumenta, o hacia el *cluster* más cercano si la densidad de datos alrededor de  $w_i$  es baja. Finalmente, para cada par de neuronas vecinas calcula las siguientes diferencias:

$$G_{i,j} = \|g_i - g_j\| \quad (5.9)$$

que son detectores de bordes entre *clusters*. Los valores o distancias de  $G_{i,j}$  son convertidos a tonos de gris de forma análoga al método U-matrix. La Figura 5.23 muestra este método para los MAO (20 × 20)-dimensionales adaptados a la distribución formada por ocho *clusters* disjuntos (a), y a la distribución formada



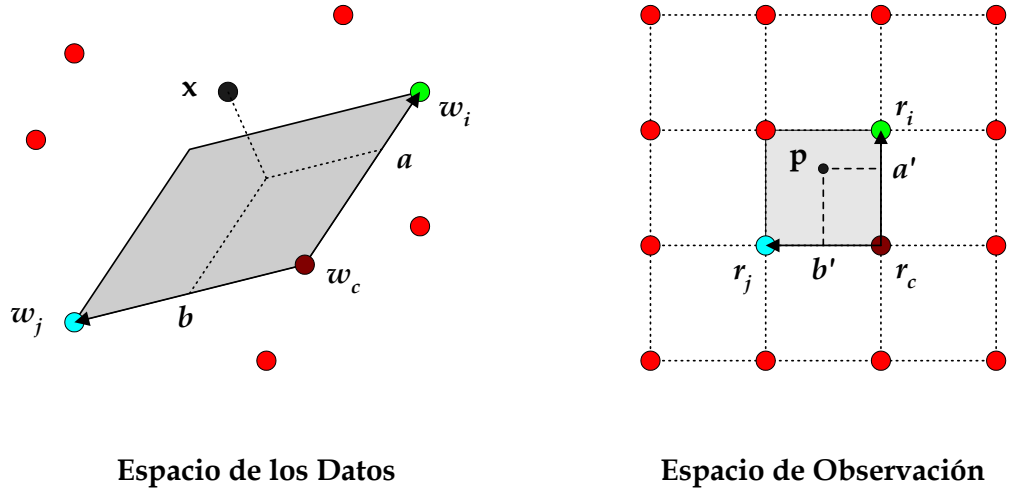


Figura 5.24: Interpolación basada en coordenadas afines de un sistema local.

por dos eslabones de una cadena (b). En ambos casos el histograma de las imágenes ha sido ecualizado. La idea de combinar información de datos y distancias también es aprovechada en el nuevo método presentado en la Sección 6.2, el cual está basado en la superposición de una matriz de distancias y un histograma de datos.

#### 5.4.4. Interpolación basada en transformaciones afines

Los métodos básicos de interpolación presentados en la Sección 5.2.3 sólo tienen en cuenta la UMS para calcular la proyección de un dato sobre el mapa. En [GR93, GR95a, GR95b] se describe un método de interpolación basado en coordenadas afines de sistemas locales, el cual utiliza tres neuronas para calcular la proyección del dato, cuando el MAO es bidimensional. En concreto, utiliza la UMS  $\eta_c$ , y las dos neuronas  $\eta_i$  y  $\eta_j$ , cuyos vectores de referencia se encuentren lo más cerca del dato de entrada  $x$ , con la restricción de que estas tres neuronas estén ubicadas en posiciones adyacentes en la rejilla regular del mapa. La idea principal del método, véase la Figura 5.24, consiste en proyectar el dato original sobre el sistema local (plano), en el espacio de los datos, cuyos vectores base son  $w_i - w_c$  y  $w_j - w_c$ , y cuyo origen es  $w_c$ . Posteriormente, realiza una extrapolación lineal de las coordenadas afines calculadas ( $a$  y  $b$ ) para determinar la posición final del dato proyectado en el mapa  $p$  (definido mediante las coordenadas  $a'$  y  $b'$ ). El sistema local de referencia es formado por las posiciones de las tres neuronas de

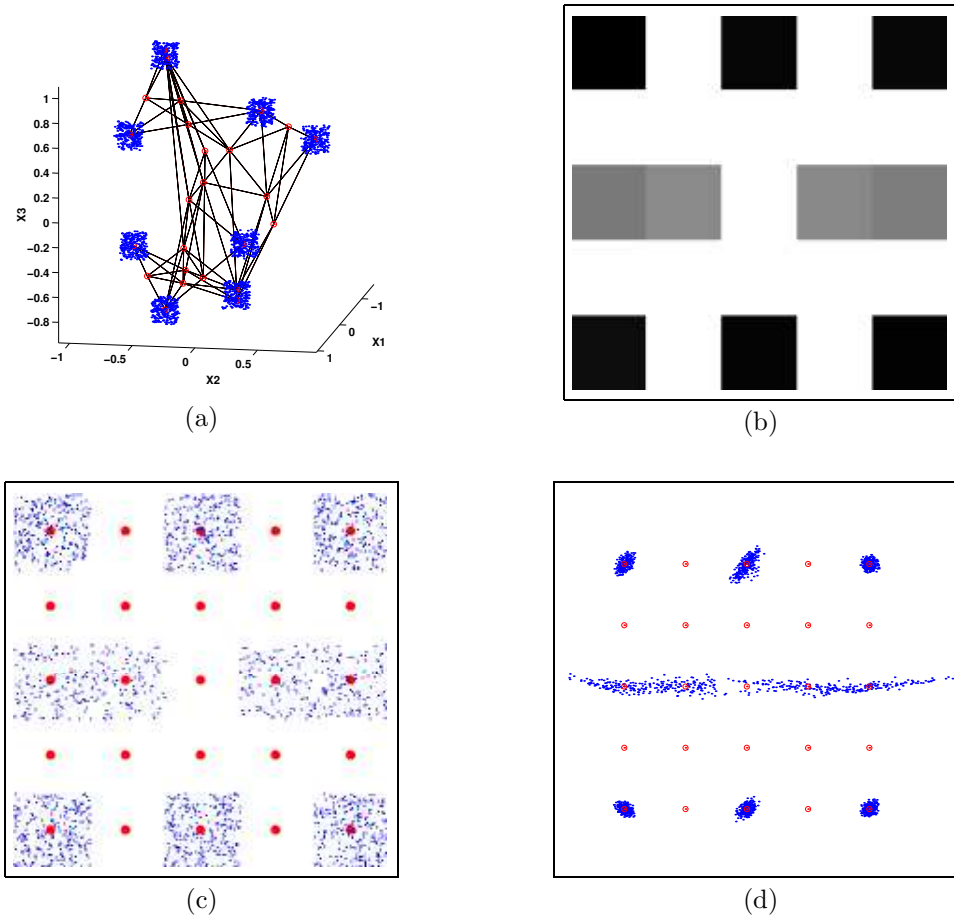


Figura 5.25: Interpolación basada en transformaciones afines. MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por ocho *clusters* (a). Histograma de datos (b). Interpolación aleatoria (c). Proyección resultante del método de interpolación basado en transformaciones afines (d).

forma análoga al del espacio de datos.

La Figura 5.25 ilustra un ejemplo del método para un MAO de pequeño tamaño  $((5 \times 5)$ -dimensional) adaptado a la distribución formada por ocho *clusters* disjuntos (a). El histograma de datos (b) y la interpolación aleatoria (c) aportan la misma información que la proyección resultante del método (d), la cual ilustra la ubicación de los *clusters* en el mapa y las neuronas que los representan.

Uno de los problemas principales de esta técnica es su limitada utilidad práctica. Al tener en cuenta únicamente tres neuronas, los datos son proyectados generalmente cerca de la neurona ganadora. De esta manera, el método aporta prácticamente la misma información que un histograma de datos o una interpolación aleatoria. Cuando el MAO es de tamaño medio o grande esto puede ser

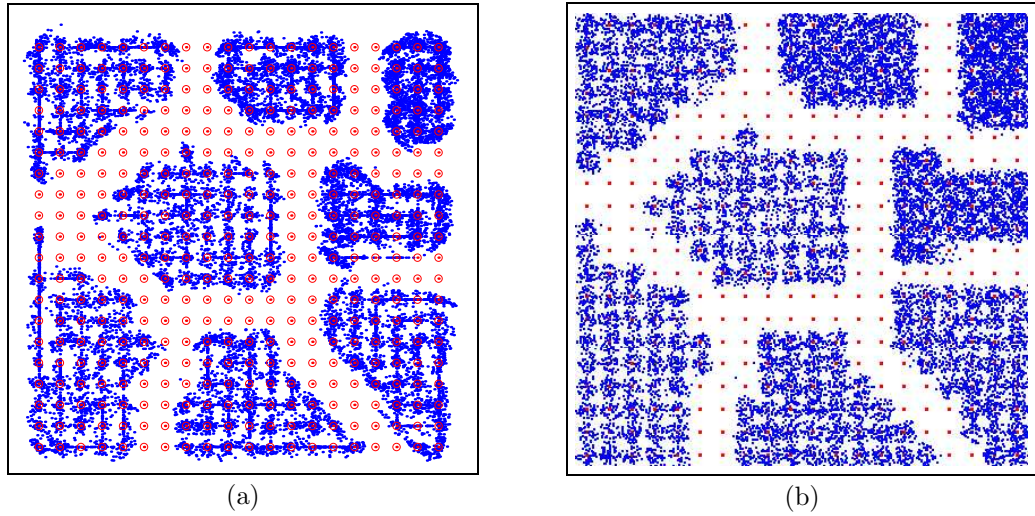


Figura 5.26: Proyección generada por el método de interpolación basado en transformaciones afines para el MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos (a). Proyección generada por la interpolación aleatoria basada en el error de cuantización para el mismo MAO (b).

más evidente. Como las neuronas suelen representar sólo una parte de una clase o *cluster* la estructura fina de los datos alrededor de una neurona no suele ser relevante (salvo en casos muy especiales como bordes entre *clusters* cuyos vectores de referencia se encuentran cercanos, o *clusters* representados por muy pocas neuronas). La Figura 5.26 muestra la proyección del método para el MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos (a), y la proyección generada por la interpolación aleatoria basada en el error de cuantización para el mismo MAO (b).

En cualquier caso, cuando el MAO es pequeño la interpolación puede ser una alternativa viable frente a la mayoría de métodos de visualización, los cuales suelen tener problemas cuando el MAO contiene pocas neuronas. Obsérvese que con pocos vectores de referencia el MAO no modelará la distribución con suficiente precisión, requiriendo un análisis de datos en vez de vectores de referencia. En estos casos los *clusters* quedarán representados o modelados por pocas neuronas, con lo cual la forma de los datos proyectados puede aportar información útil sobre la distribución, siempre que el método de proyección o interpolación sea preciso y aporte descripciones adecuadas de los datos.

En cuanto a la calidad y exactitud del método a la hora de representar gráficamente las distribuciones hay que destacar que no fue diseñado para visualizar

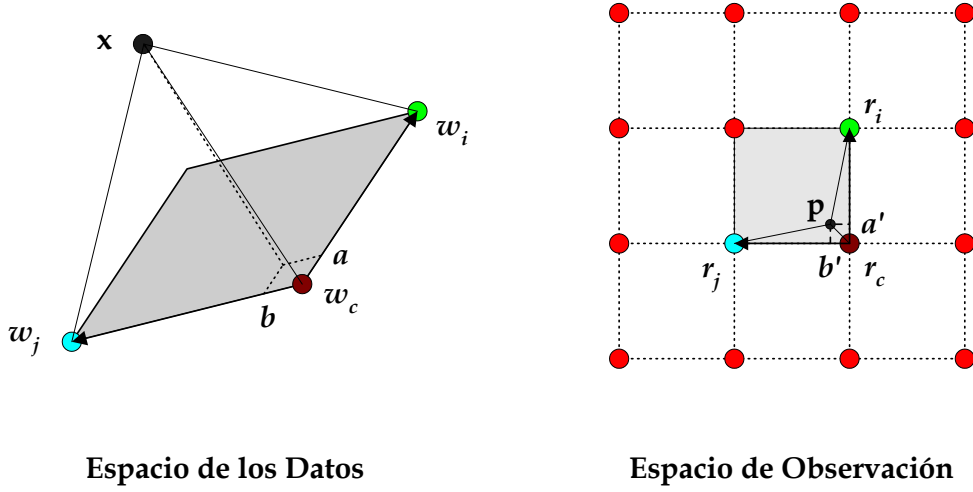


Figura 5.27: La interpolación basada en coordenadas afines de un sistema local no preserva adecuadamente las distancias cuando el dato a proyectar se encuentra alejado del plano formado por los tres vectores de referencia asociados al dato en los que la proyección está basada.

datos, sino para aumentar la velocidad de búsqueda de la UMS y acelerar el entrenamiento de los MAO. De esta manera, presenta deficiencias significativas a la hora de visualizar distribuciones que limitan su uso incluso para mapas pequeños:

- Existen situaciones en las que los tres vectores de referencia o las tres neuronas no forman una base, es decir, son linealmente dependientes. Este problema puede ser tratado en el propio algoritmo, escogiendo adecuadamente las neuronas en torno a la ganadora.
- Al estar basado especialmente en la UMS la proyección es discontinua, ya que dos patrones muy próximos con distintas UMS pueden ser proyectados a posiciones muy diferentes en el mapa.
- Las proyecciones pueden aparecer “invertidas” (véase la Figura 5.28).
- El método no es robusto ante situaciones en las que el MAO no se ha adaptado adecuadamente a la distribución, tanto de manera ordenada como con un error de cuantización bajo.
- A medida que el dato a proyectar se aleja del plano formado por los tres vectores de referencia seleccionados las distancias desde el dato a éstos tienden a igualarse, mientras que las distancias entre el punto proyectado y las

neuronas en el mapa pueden resultar muy diferentes. La Figura 5.27 ilustra este problema. El dato  $x$  es proyectado cerca de la neurona ganadora, donde la distancia  $\|p - r_c\|$  es pequeña en comparación con  $\|p - r_i\|$  y  $\|p - r_j\|$ . Sin embargo, las distancias originales desde  $x$  a sus tres vectores de referencia más cercanos (vecinos del ganador en el mapa) son parecidas  $\|x - w_c\| \simeq \|x - w_i\| \simeq \|x - w_j\|$ .

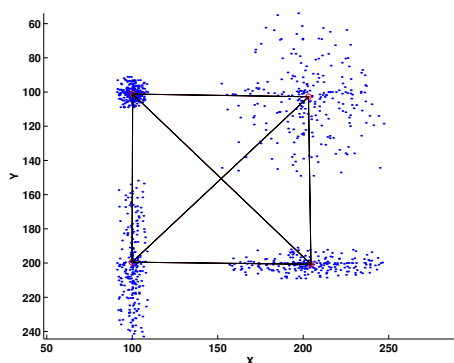
La Figura 5.28 muestra como el método tiene problemas incluso cuando el espacio de los datos es bidimensional. Con los ejemplos se persigue averiguar si el método es capaz de revelar la estructura local de los datos alrededor de cada vector de referencia. Para ello dos MAO ( $2 \times 2$ )-dimensionales han sido entrenados con dos distribuciones bidimensionales: cuatro clases con igual tamaño y diferente forma (a), y una circunferencia (b). Las respectivas proyecciones son mostradas en (c) y (d). En ambos casos el método presenta deficiencias, incluso con esas distribuciones “sencillas”. Las proyecciones de las diferentes clases asociadas a cada vector de referencia aparecen “invertidas”. Por otro lado, es posible apreciar claramente la discontinuidad del método al proyectar la circunferencia.

La Sección 6.1 propone un nuevo método de interpolación [RDG<sup>+</sup>02, RG03] que puede considerar un número mayor de neuronas para calcular la proyección. El resultado es un algoritmo que presenta varias propiedades opuestas a las del método de interpolación presentado en este apartado, resolviendo la mayoría de problemas asociados a éste. La Sección 6.1 presenta más ejemplos de la proyección del método de interpolación basado en transformaciones afines, con el objetivo de compararlo con el nuevo método.

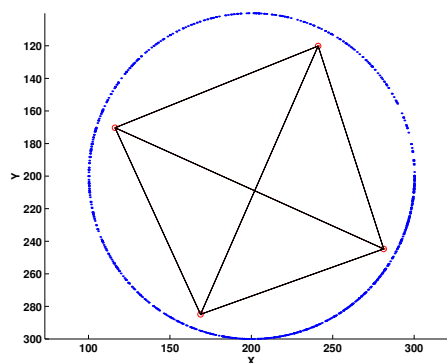
## 5.5. Proyecciones y Coloración

Este apartado describe varios métodos sofisticados de visualización de MAO previamente descritos en la literatura. La idea principal consiste en generar imágenes en las que cada *cluster* de la distribución aparezca con un color diferente. De esta manera, no sólo resulta intuitivo identificar cada *cluster* y los bordes entre éstos, sino que además se facilita el etiquetado y el análisis trayectorias o secuencias de patrones de entrada. Existen dos estrategias principales a la hora de crear estas visualizaciones:

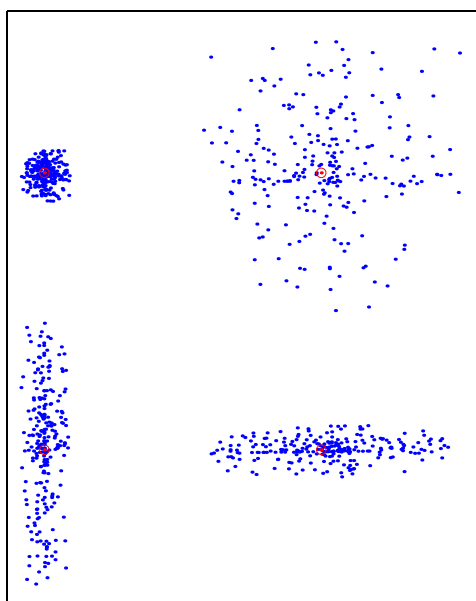
- *Clustering* del MAO. Los vectores de referencia del MAO son agrupados en varias clases a las que se asignan colores diferentes.



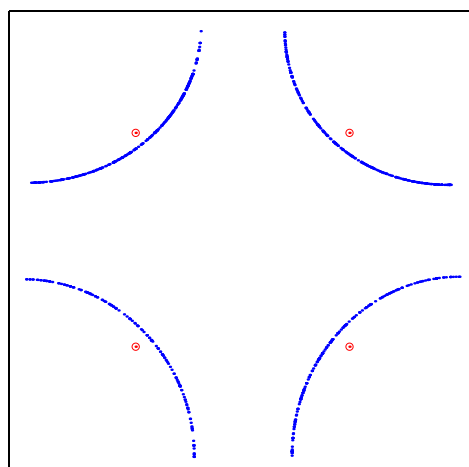
(a)



(b)



(c)



(d)

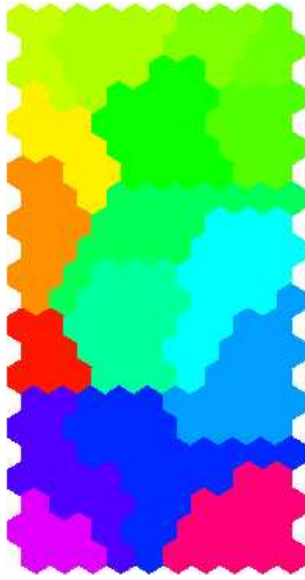
Figura 5.28: Deficiencias de la interpolación basada en transformaciones afines. Dos MAO ( $2 \times 2$ )-dimensionales son adaptados a dos distribuciones bidimensionales: cuatro clases con igual tamaño y diferente forma (a), y una circunferencia (b). Las respectivas proyecciones son mostradas en (c) y (d). En ambos casos las proyecciones aparecen “invertidas”. La discontinuidad de la proyección es clara al proyectar la circunferencia.

- Encadenado de proyecciones mediante códigos de colores. La información de dos métodos de proyección es combinada mediante un código de colores (un elemento proyectado recibe el mismo color en ambas proyecciones, aunque la asignación de dicho color sólo depende de la posición del elemento en una proyección). Esta estrategia ha motivado la creación de varios nuevos algoritmos de proyección o “reorganización” de las neuronas en el espacio de observación de los MAO. A partir de éstos es posible generar mapas donde las diferencias entre los colores de los *clusters* corresponden de alguna manera a las distancias entre dichos *clusters*. La Sección 6.3 propone un algoritmo rápido para generar estas imágenes cuya estructura es muy similar a la del algoritmo de aprendizaje secuencial del MAO.

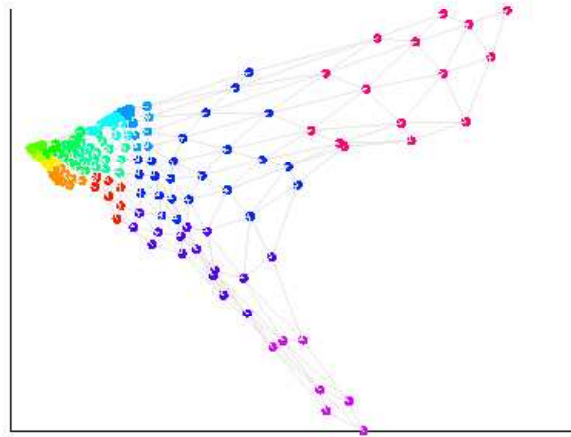
### 5.5.1. *Clustering* del MAO

El *clustering* es una de las aplicaciones más importantes de los MAO en la minería de datos [Ves02]. Los propios vectores de referencia pueden ser agrupados en diferentes clases aplicando cualquier algoritmo de CV tradicional (K-medias, algoritmos jerárquicos, etc.) [VA00]. También es posible emplear otros enfoques más relacionados con la arquitectura de los MAO, como estrategias que tienen en cuenta sus relaciones de vecindad [Mur95] o estrategias basadas en las matrices de distancias [Ult99, VLM99, VH02]. Estos enfoques resultan ser considerablemente más rápidos que realizar una CV de los datos directamente, y la correspondencia entre los resultados de ambos enfoques es buena [Ves02]. Una vez hallados los diferentes *clusters* en la rejilla regular del MAO se asigna un color diferente a cada uno de ellos, apareciendo un mapa coloreado que aporta información acerca de su estructura. En la práctica, al igual que en la visualización, cuando el objetivo es encontrar los *clusters* naturales de la distribución de datos el número de vectores de referencia del MAO debe ser mucho mayor que el número esperado de *clusters* [Ult99].

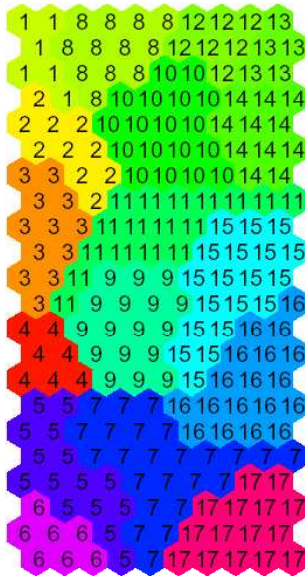
La Figura 5.29 muestra un ejemplo de *clustering* del MAO tomado de [Ves02]. El MAO utilizado (cuya topología es hexagonal) es  $(22 \times 10)$ -dimensional, mientras que los datos de entrenamiento pertenecen a varias medidas asociadas a un computador en un entorno de red: tasa de utilización de la unidad central de proceso, volumen de tráfico en la red, etc. En (a) una primera clasificación de los vectores de referencia genera 17 *clusters*, los cuales son representados mediante



(a)



(b)



(c)

Figura 5.29: *Clustering* automático de un MAO. Los diferentes *clusters* hallados aparecen con diferentes colores (a). Proyección de los vectores de referencia del MAO según las dos primeras componentes principales de los datos (b). En (c) un algoritmo jerárquico aglomerativo construye un dendrograma en función de los *clusters* hallados. Tomado de [Ves02].



diferentes colores (en este caso el algoritmo de CV está basado en mínimos locales de la matriz de distancias). La proyección de los vectores de referencia según las dos primeras componentes principales de los datos es mostrada en (b), donde también se ilustran las conexiones entre los vecinos en el MAO y donde cada vector de referencia aparece con el color que le corresponde en el MAO coloreado. La proyección de los vectores de referencia muestra la forma de los datos mejor que la rejilla regular del MAO, sin embargo, es más difícil apreciar los detalles en las áreas densas si no se dispone de una interfaz interactiva para poder realizar un *zoom*. Por último, en función de una distancia entre *clusters*, éstos pueden ser fusionados mediante un método jerárquico aglomerativo para generar estructuras de mayor tamaño, apareciendo un clásico dendograma (c) (donde los colores de los “*super-clusters*” son calculados como promedios de los de sus “*sub-clusters*”).

### 5.5.2. Encadenado de proyecciones mediante códigos de colores

Los MAO deben ser analizados por varios métodos de visualización. Los datos generalmente contienen tanta información que resulta imposible mostrarla en una sola imagen (véase la Sección 6.2.1). El número de “dimensiones visuales” determina los tipos de información que pueden ser incluidos en una visualización eficientemente. Las dimensiones visuales típicas incluyen posición, tamaño y color (o textura). La forma o el movimiento son otras dimensiones visuales aunque su uso es más problemático. En entornos de realidad virtual otros sentidos como el oído o el tacto pueden intervenir también [Ves99].

Dado el número limitado de dimensiones visuales el analista debe hacer uso de varias visualizaciones, por lo que es necesario encontrar una forma de asociar las diferentes imágenes obtenidas. En la literatura, dicha asociación se denomina encadenado, y generalmente es generado entre dos proyecciones distintas de los vectores de referencia del MAO al espacio de observación. En [Ves02] el autor distingue cuatro tipos de encadenado: por posición, por color, utilizando líneas explícitamente y por movimiento. En el caso del MAO el encadenado suele ser por posición, ya que la mayoría de visualizaciones están basadas en la rejilla regular, y por tanto utilizan la misma proyección de los vectores de referencia (las posiciones de las neuronas en el mapa). El método de superposición de imágenes (véase la Sección 6.2) puede ser visto como un encadenado por posición, donde

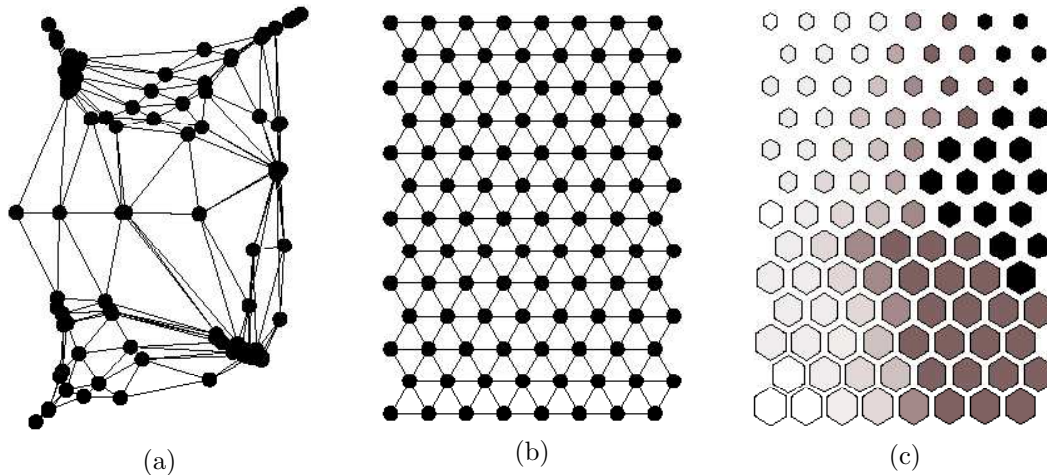


Figura 5.30: Encadenado de dos proyecciones de los mismos elementos mediante tamaño e intensidad luminosa. Primera proyección no-lineal de un conjunto de vectores de referencia (a), y segunda proyección de un MAO (b). A cada elemento proyectado se le asigna un tamaño y nivel de gris en función de sus coordenadas en (a). Posteriormente, cada neurona es visualizada mediante un hexágono cuyo tamaño y nivel de gris corresponde al asignado a su vector de referencia en la primera proyección (c). Tomado de [Him00].

dos imágenes son colocadas una encima de la otra.

En este apartado, sin embargo, el objetivo es encadenar dos proyecciones distintas de los vectores de referencia de un MAO. En otras palabras, se trata de relacionar dos proyecciones diferentes de una serie de centroides o vectores de referencia de un método de CV. El objetivo es combinar dos proyecciones de datos de dimensión elevada para obtener más información en las visualizaciones bidimensionales, ya que diferentes proyecciones aportan diferentes relaciones en los datos [Him98]. Una de las proyecciones corresponde a la estructura regular de la rejilla de neuronas del MAO, mientras la segunda procede de otro método de proyección lineal, no-lineal o de otras estrategias diseñadas para conseguir una organización diferente de las neuronas en el espacio de observación.

Al contar con dos proyecciones distintas es necesario utilizar un encadenado no basado en la posición. La alternativa más utilizada (y posiblemente más eficaz) consiste en realizar un encadenado basado en códigos de colores, aunque es posible utilizar otras características (tamaño, intensidad luminosa, forma, orientación, etc.). La Figura 5.30 muestra un primer ejemplo del encadenado de proyecciones utilizando como características el tamaño e intensidad luminosa. La figura muestra una primera proyección (obtenida a partir del método de contracción de

Himberg [Him00], véase la Sección 5.5.2.3) de un conjunto de vectores de referencia (a), y una segunda del mismo conjunto de vectores de referencia (neuronas) que genera un MAO (b). En función de sus coordenadas en (a), a cada vector proyectado se le asigna un determinado grado de intensidad luminosa (según el eje horizontal) y un tamaño (según el eje vertical). Posteriormente, las neuronas de la segunda proyección pueden ser descritas mediante elementos (hexágonos) cuyo tamaño y nivel de gris corresponden a los asignados en la primera proyección a sus vectores de referencia (c). De esta manera, es posible detectar varios *clusters* identificando zonas de poca variación de los hexágonos.

La idea del encadenado mediante códigos de colores primero aparece en [KK98], donde es utilizada para visualizar la estructura de *clusters* en los MAO. Se trata de asignar un color a cada neurona o centroide proyectado en función de sus coordenadas en el espacio de observación, para posteriormente representar con el mismo color al centroide o neurona de la segunda proyección. De esta manera, resulta sencillo visualizar la conexión entre ambas proyecciones. Las visualizaciones pueden ser mejoradas introduciendo las conexiones entre vectores de referencia vecinos [Him98].

La Figura 5.31 ilustra el encadenado de dos proyecciones mediante colores. En (a) las neuronas (y por tanto sus vectores de referencia) de un MAO han sido asignados un color en función de su posición en la rejilla regular. En este caso el código de colores utilizado corresponde a una sección (plano  $B = 255 - G$ ) del cubo RGB (b). En concreto, una neurona situada en la posición  $(x, y)$  recibe el correspondiente color  $255(1 - y, 1 - x, x)$ , donde las coordenadas son escaladas al intervalo  $[0, 1]$  y 255 es el máximo valor de intensidad luminosa. En cualquier caso, es importante tener en cuenta que el uso de una sección del cubo RGB es una solución claramente heurística, al no tener en cuenta factores psico-visuales [Ves99]. Posteriormente, en (c) cada vector proyectado (según un segundo método de proyección) es ilustrado con su correspondiente color en el mapa coloreado, donde los vectores de referencia vecinos han sido conectados mediante segmentos. Por otro lado, el encadenado también puede ser realizado en sentido contrario. Los vectores proyectados en (c) pueden recibir un color según el mismo código lineal, para posteriormente colorear las neuronas del MAO (d), donde áreas o neuronas con matices similares corresponden, en la mayoría de casos, a vectores de referencia cercanos en el espacio de los datos.

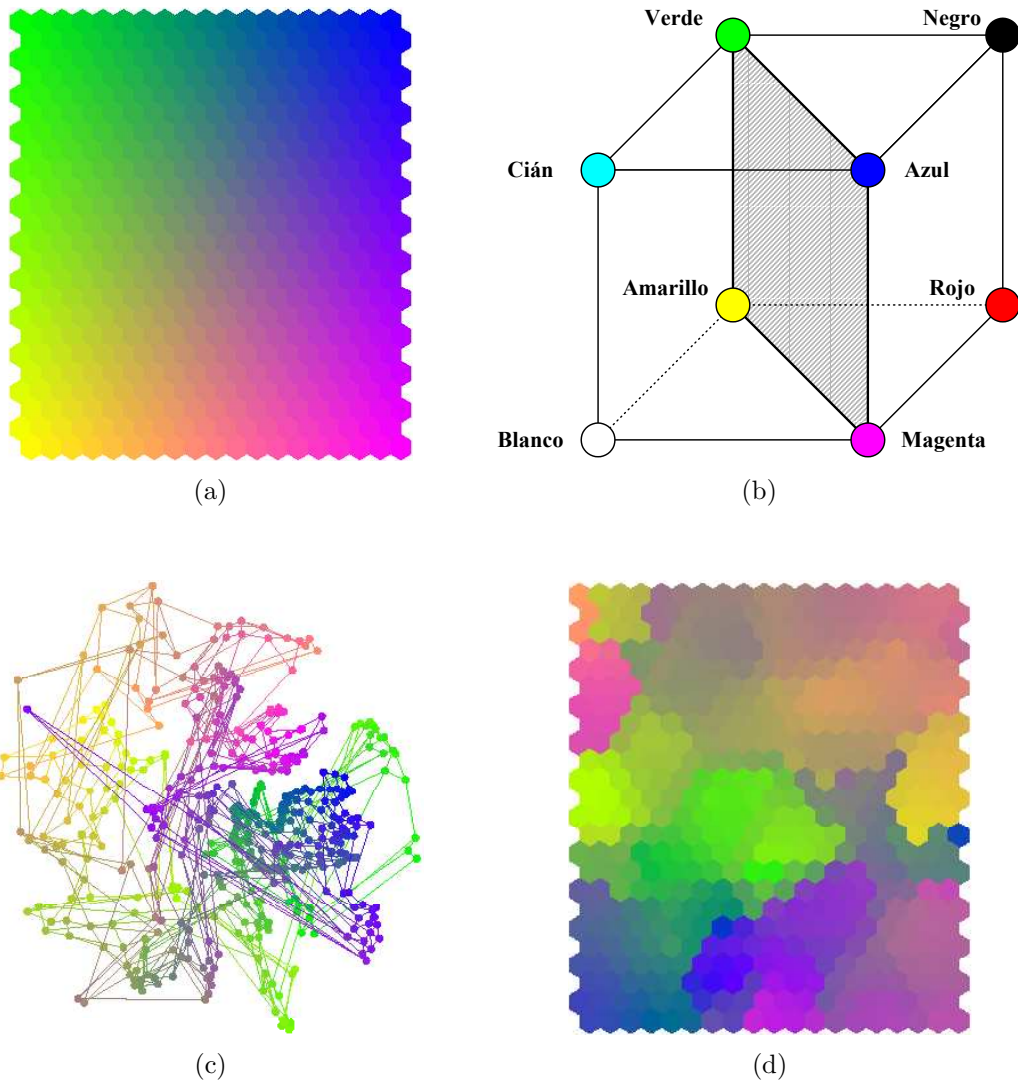


Figura 5.31: Encadenado de dos proyecciones mediante un código de colores lineal. Las neuronas (y sus vectores de referencia asociados) en (a) reciben un color en función de su posición, donde el código de colores lineal utilizado es una sección del cubo RGB (b). Los mismos vectores de referencia toman ese mismo color en una proyección diferente a un plano bidimensional (c). En (d) el proceso es realizado en sentido contrario. Tomado de [Ves99], excepto la imagen (b).

En esta tesis, en la cual los MAO tratados son generalmente bidimensionales, el rectángulo (o cuadrado) que define el código de colores es reescalado para ajustarlo al rectángulo definido por los valores mínimos y máximos de las proyecciones:  $\min_i\{r_{i,x}\}$ ,  $\max_i\{r_{i,x}\}$ ,  $\min_i\{r_{i,y}\}$  y  $\max_i\{r_{i,y}\}$ , para los límites izquierdo, derecho, abajo y arriba, respectivamente, donde  $r_i = (r_{i,x}, r_{i,y})$  corresponde a la posición o proyección del vector de referencia  $w_i$ .

A pesar de la importancia del color en estas visualizaciones, es importante tener en cuenta que una segunda proyección diferente a la del MAO también aporta información por sí misma, donde generalmente es posible detectar diversas estructuras de los datos. Por otro lado, en estos casos y en la mayoría de situaciones en las que se realiza un encadenado mediante códigos de colores, es importante trabajar con un elevado número de vectores de referencia, ya que son éstos los que modelan la distribución y los que van a aportar los detalles necesarios para realizar un análisis efectivo de los datos.

### 5.5.2.1. MAO + proyección de Sammon

La rejilla del MAO describe las relaciones topológicas de los datos, pero no muestra las distancias entre los vectores de referencia vecinos. Aunque es posible utilizar matrices de distancias para ese propósito, suele ser habitual proyectar los vectores de referencia mediante la proyección de Sammon (véase la Sección 2.5.5.1) para obtener una visualización que intente preservar las distancias mutuas entre ellos. Esto aporta una idea sobre las distancias entre los vectores de referencia e incluso puede revelar como el MAO se dobla en el espacio de los datos. Sin embargo, suele ser difícil ver la conexión entre la proyección de Sammon de los vectores de referencia y la proyección del propio MAO.

Una solución consiste en utilizar el encadenado de ambos métodos de proyección, el cual puede ser realizado en cualquier sentido. La Figura 5.32 muestra las dos formas de encadenar las dos proyecciones. En (a) primero se asignan colores a las neuronas en la rejilla hexagonal de un MAO  $(11 \times 9)$ -dimensional para posteriormente colorear la proyección de Sammon. En (b) el proceso es realizado a la inversa, donde la proximidad de los vectores de referencia en el mapa queda reflejada en las diferencias entre los colores.

Este procedimiento es útil porque aporta una “segunda opinión” al combinar dos métodos de visualización [Him98], permitiendo realizar un análisis más completo de la estructura de *clusters* de la distribución. Por otro lado, si el MAO ha

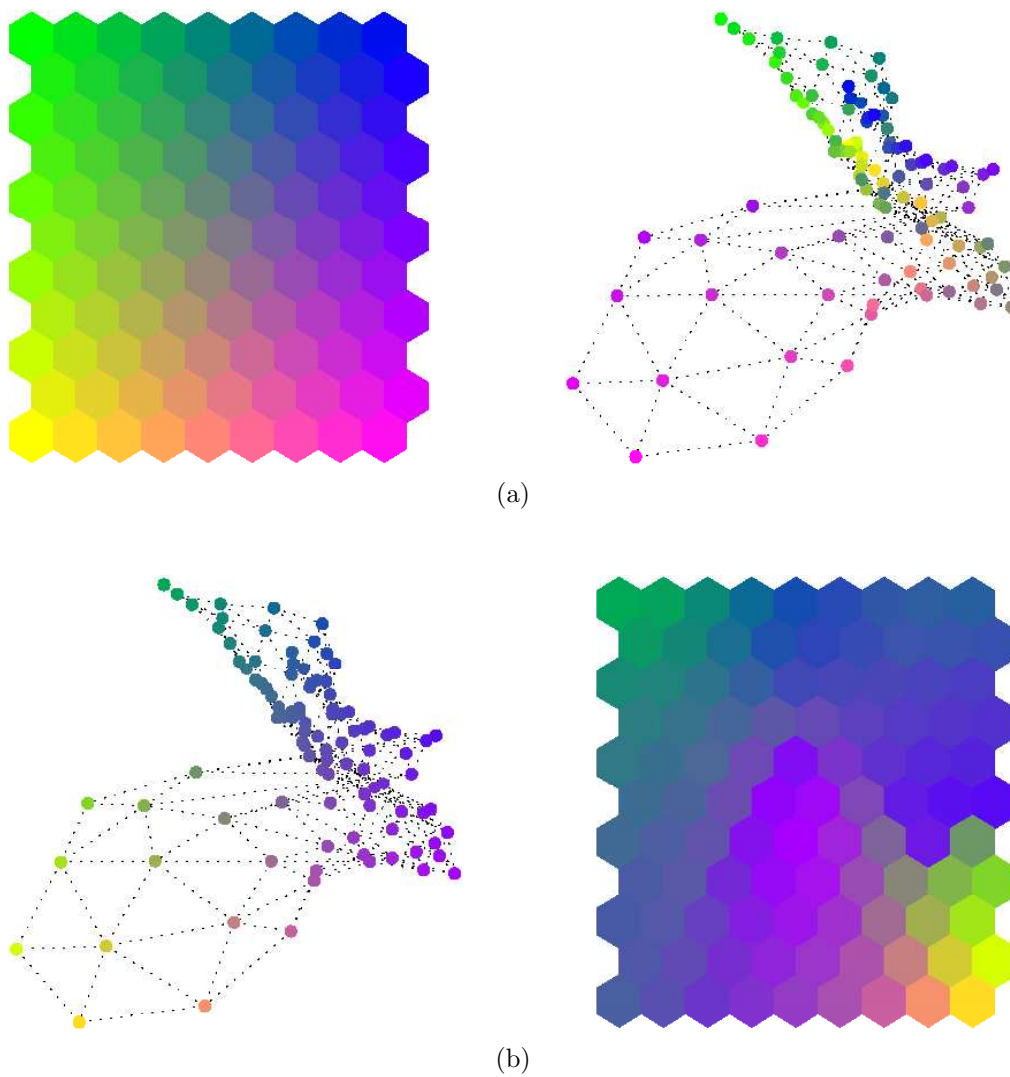


Figura 5.32: Encadenado de un MAO con la proyección de Sammon de los vectores de referencia. MAO  $\rightarrow$  Sammon (a). Sammon  $\rightarrow$  MAO (b). Tomado de [Him98].

sido entrenado con datos sobre el estado de un proceso, las trayectorias pueden ser analizadas de un modo intuitivo representando consecutivamente los colores de las neuronas ganadoras, por ejemplo, en una lista. Esto suele resultar más sencillo que analizar grafos dirigidos (véase la Sección 5.2.4). Por otro lado, puede ayudar a detectar vectores de referencia lejanos en el mapa pero cerca en el espacio de los datos, y viceversa. En definitiva, el color aporta más información a las visualizaciones, y puede ser combinado con otros métodos de análisis exploratorio para mejorarlos. Por ejemplo, los diagramas de dispersión pueden ser notablemente mejorados si cada elemento proyectado es ilustrado con el color asociado a la neurona ganadora en un MAO coloreado [Him98].

### 5.5.2.2. Diferencias perceptivas

Es de especial interés colorear las neuronas del MAO de tal forma que el resultado refleje de algún modo la estructura de *clusters* de los vectores de referencia [Him00]. Un modelo riguroso para alcanzar este objetivo es presentado en [KVK99, KVK00]. El modelo utiliza un proceso de optimización estocástico para proyectar los vectores de referencia a un plano bidimensional de tal forma que las distancias entre los vecinos sean preservadas aproximadamente, y se preserve el ordenamiento local definido por la rejilla regular. Además, la proyección puede ser restringida para encajar en el espacio del modelo de color CIELab (véase [Fai98]), de forma que las diferencias perceptivas entre los colores reflejen las relaciones entre las distancias locales en la estructura de *clusters* lo más fielmente posible, y donde el ordenamiento topológico global del MAO refleje estructuras de mayor tamaño en los datos [Him00].

Se trata de una variante de los métodos de EMD. Realiza una proyección de los vectores de referencia de un MAO, con una restricción añadida para preservar el orden topológico de las neuronas, y una nueva restricción para que la proyección sea realizada sobre una sección (de brillo o luminosidad constante) del espacio de color CIELab. En este caso, se evita usar el modelo RGB por dos razones: en primer lugar, depende del dispositivo de visualización; por otro lado, las diferencias perceptivas entre los colores en diferentes partes del espacio de color no son uniformes. Es decir, la diferencia perceptiva entre un par de colores que tienen la misma distancia Euclídea en el espacio de color RGB varía mucho dependiendo de la posición del par en el espacio de color. Afortunadamente, existe un grupo de espacios de color (los uniformes) definidos de tal forma que

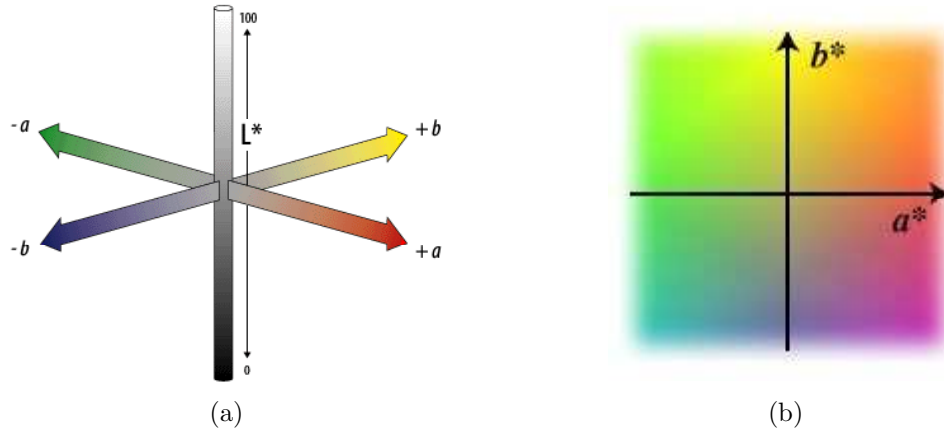


Figura 5.33: Espacio de color CIE Lab (a), tomado de [URLh]. Sección del espacio (b), tomado de [URLi].

las diferencias perceptivas en color correspondan a las distancias Euclídeas en el espacio, lo mejor posible. Uno de estos espacios es el CIE Lab, recomendado por la *Commission Internationale de L'Éclairage*. Este modelo está basado en la teoría de colores oponentes de Ewald Hering, en la que los estímulos de color en la retina son traducidos a diferencias entre claro y oscuro, rojo y verde, y azul y amarillo [URLh] (véase la Figura 5.33).

De esta manera, el método define una nueva reorganización (proyección) de los vectores de referencia, mientras el código de colores está basado en el modelo uniforme CIE Lab, en vez de en el RGB. Formalmente, se trata de un método de EMD con restricciones, que minimiza la siguiente función de error:

$$E = E_1 + \lambda_2 E_2 + \lambda_3 E_3 \quad (5.10)$$

donde  $\lambda_2$  y  $\lambda_3$  son parámetros que son escogidos “adecuadamente”.  $E_1$  es un clásico error de EMD para preservar las distancias entre los vectores de referencia vecinos:

$$E_1 = \sum_i \sum_{j \in \tilde{\Xi}_i} (d_{i,j} - d'_{i,j})^2 \quad (5.11)$$

siendo  $d_{i,j} = \|w_i - w_j\|$ ,  $d'_{i,j} = \|w'_i - w'_j\|$ , donde  $w'_i$  es la proyección de  $w_i$  a un espacio de baja dimensión, y  $\tilde{\Xi}_i$  es el conjunto de neuronas vecinas a la  $i$  (sin incluirla a ella misma). El error  $E_2$  es añadido para conseguir un ordenamiento



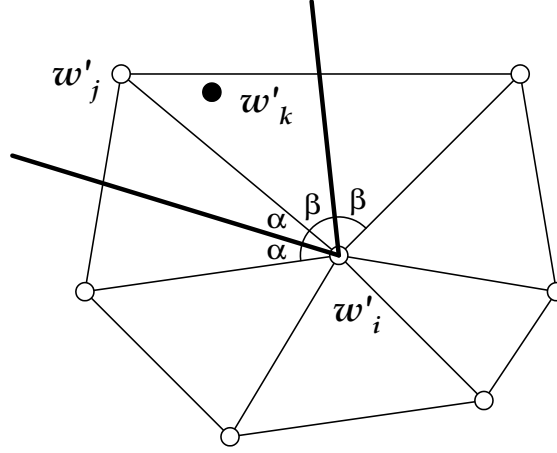


Figura 5.34: Definición de ordenamiento de una proyección. Las proyecciones de los vectores de referencia vecinos a la neurona  $\eta_i$  son ilustrados mediante círculos blancos, donde los vecinos están conectados mediante líneas delgadas. Si un vector proyectado no-vecino  $w'_k$  pertenece al “sector” de  $w'_j$ , ilustrado mediante líneas gruesas, y está más cerca de  $w'_i$  que de  $w'_j$ , entonces está demasiado cerca de  $w'_i$ , y la proyección no está ordenada. El sector consiste en los puntos para los cuales el ángulo desde  $w'_j$ , visto desde  $w'_i$ , es más pequeño que el visto desde el resto de vecinos del punto central  $w'_i$  [KVK00].

de los vectores proyectados:

$$E_2 = \sum_{i,j \text{ vecinos}} \sum_{k \in I_{i,j}} (d'_{i,j} - d'_{i,k})^2 \quad (5.12)$$

donde  $I_{i,j}$  define un conjunto de índices que no satisfacen una restricción de ordenamiento para el par de vecinos  $(i, j)$ . La idea consiste, *grosso modo*, en evitar que un vector proyectado no vecino del  $w_i$  entre dentro del cierre convexo formado por las proyecciones pertenecientes a  $\tilde{\Xi}_i$ . La definición detallada es descrita e ilustrada en la Figura 5.34. Finalmente,  $E_3$  es utilizado para escalar las distancias adecuadamente dentro del espacio CIELab:

$$E_3 = \sum_i \|\mathcal{K}\bar{w}'_i - w'_i\|^2 \quad (5.13)$$

donde  $\mathcal{K}$  es una constante tal que  $\bar{w}'_i$  es el punto en el espacio disponible más cerca de  $w'_i/\mathcal{K}$  (véanse [KVK99, KVK00]).

La Figura 5.35 muestra un ejemplo de esta proyección y coloración para el “mapa fonético” cuya U-matrix es ilustrada en la Figura 5.15. La proyección del mapa fonético a un espacio bidimensional (sin considerar el error  $E_3$ ) es ilustrada

## 5. MÉTODOS TRADICIONALES DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS

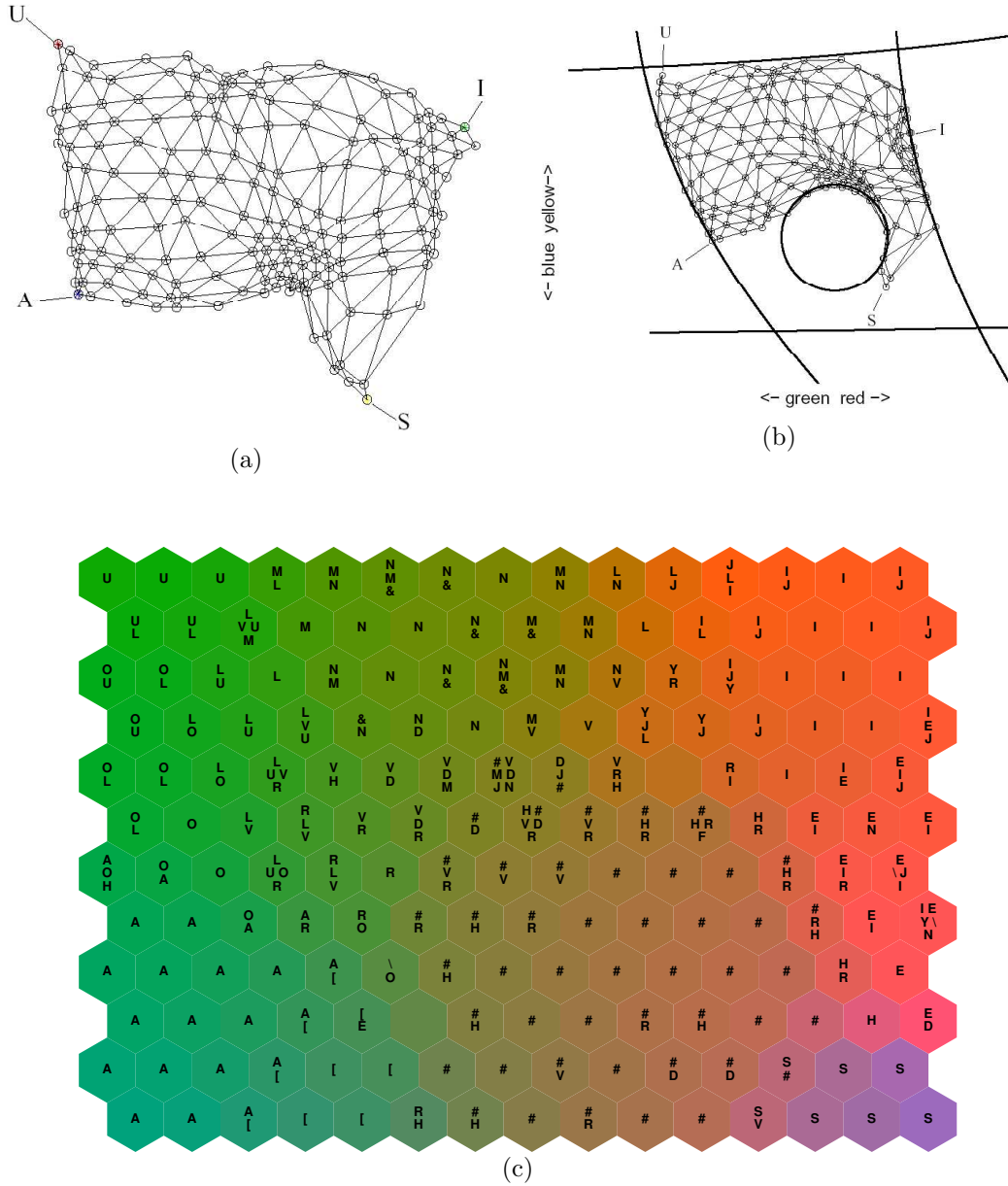


Figura 5.35: Proyecciones del método basado en diferencias perceptivas. Proyección del mapa fonético a un espacio bidimensional sin tener en cuenta el error  $E_3$  (a). Incorporando  $E_3$  los vectores de referencia son proyectados a una sección de luminosidad constante del espacio CIELab (b). Tomados de [KVK00]. A través del encadenado es posible construir un mapa fonético coloreado (c). Tomado de [URLj].

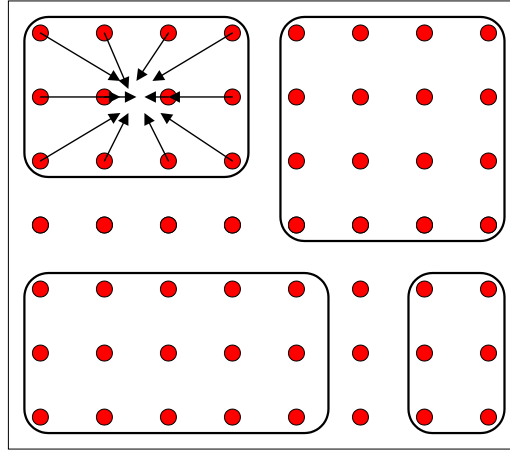


Figura 5.36: Estrategia de los métodos de contracción. Las neuronas de cada *cluster* son atraídas entre sí.

en (a). Nótese que se respeta el orden hexagonal del MAO original. Incorporando  $E_3$  los vectores de referencia son proyectados a una sección de luminosidad constante del espacio CIELab (b). Las líneas gruesas delimitan la región representable por un tubo CRT, y el círculo en el centro encierra colores demasiado poco saturados (grises). A través del encadenado con la proyección original del MAO adaptado a los datos de habla, es posible construir un mapa fonético coloreado donde las diferencias perceptivas entre los colores corresponden más fielmente a las distancias entre los vectores de referencia (c). Obsérvese la relación entre este mapa coloreado y la U-matrix. En este caso, los *clusters* no sólo aparecen con un color diferente, sino que las diferencias entre dichos colores dan una idea de la distancia relativa entre los *clusters*.

### 5.5.2.3. Método de contracción de Himberg

Este apartado describe una nueva alternativa para crear una proyección o reorganización de las neuronas (vectores de referencia proyectados) sobre el espacio de observación de un MAO. Se trata de una estrategia importante en esta tesis, ya que el algoritmo de agrupación de neuronas (véase la Sección 6.3) está basado en ella. La idea fundamental consiste en modificar las posiciones de las neuronas en el espacio de observación continuo, para atraer aquellas cuyos vectores de referencia se hallan próximos en el espacio de los datos. Es decir, aproxima aquellas neuronas que representan a un mismo *cluster*. Esta idea es ilustrada en la Figura

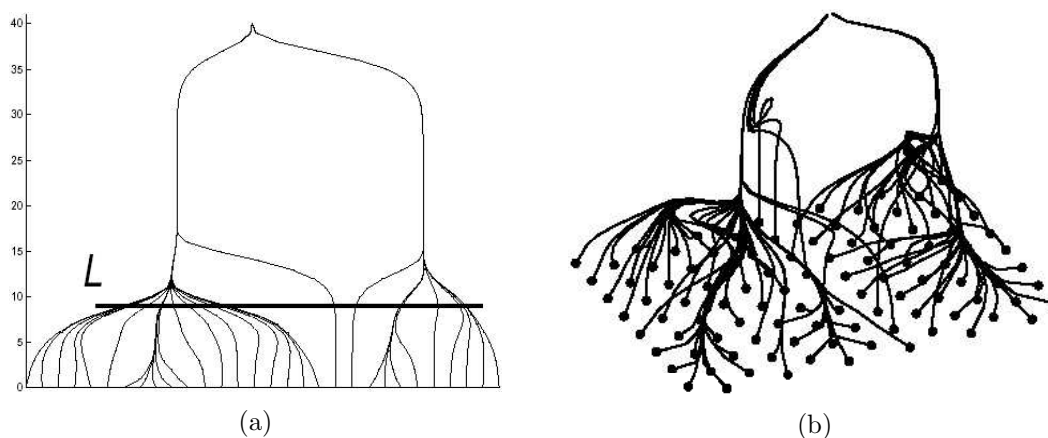


Figura 5.37: Proceso de contracción de un MAO unidimensional (a), y otro bidimensional (b). Los ejes verticales están relacionados con el número de iteraciones llevadas a cabo por los algoritmos de contracción. Tomado de [Him00].

5.36. El proceso genera una nueva proyección no-lineal de los vectores de referencia que posteriormente puede ser encadenada con la proyección original del MAO mediante un código de colores. El resultado es similar al método de diferencias perceptivas (véase la Sección 5.5.2.2), donde diferentes *clusters* aparecen con diferentes colores y las diferencias entre los colores de neuronas vecinas corresponden aproximadamente a la distancia Euclídea entre sus vectores de referencia. Por otro lado, el proceso de contracción también genera visualizaciones en forma de árboles jerárquicos o dendogramas. La Figura 5.37 ilustra el proceso de contracción para un MAO unidimensional (a) y otro bidimensional (b). Nótese que los árboles pueden ser cortados por un nivel  $L$  para detectar *clusters* automáticamente y para asignar un color a cada neurona en función de sus coordenadas en dicho nivel.

La idea está muy relacionada con los métodos de *clustering* basados en un principio de contracción o modelo gravitacional [Wri77, Kun99, Dav02, GDN03], los cuales asignan masa a cada dato y aplican el modelo gravitacional para acercarlos progresivamente, lo cual automáticamente determina una serie de *clusters*. Por ejemplo, el modelo propuesto por Wright [Wri77] es un algoritmo jerárquico aglomerativo donde las fuerzas gravitacionales son utilizadas como mecanismos para fusionar partículas (datos) hasta que sólo quede una partícula (*cluster*) en el sistema. Algunas de las propiedades más importantes de la simulación de Wright son:

- Para determinar la posición de una partícula intervienen todas las del conjunto de datos.
- Cuando dos partículas están lo suficientemente cerca como para fusionarlas, una se elimina y la masa de la segunda se incrementa con la de la eliminada.
- Un parámetro del sistema determina la distancia máxima que puede desplazarse una partícula en cada iteración del algoritmo.
- La simulación termina cuando sólo queda una partícula en el sistema.

Con respecto a la visualización de MAO, en [Him00] se describe un modelo de contracción, sencillo de implementar, para analizar la estructura de *clusters* de los vectores de referencia. Comparte varias características con el modelo gravitacional de Wright, aunque no está basado estrictamente en un modelo físico. En primer lugar, las distancias entre los vectores de referencia  $w_i$  de un MAO son calculadas y transformadas adecuadamente a similitudes. Por otro lado, las neuronas son organizadas según su configuración en la rejilla regular del MAO ( $r_i$ ), y esas posiciones van a servir como punto de partida del proceso de contracción. Cada iteración del algoritmo calcula unas nuevas coordenadas para cada neurona, en concreto, medias ponderadas de las posiciones de las neuronas en la iteración anterior. Los pesos corresponden a las similitudes entre los vectores de referencia de las neuronas (normalizados para que sumen uno). Si la operación de promediado es repetida, eventualmente todas las neuronas convergen a un punto. Las trazas de estos puntos pueden ser visualizadas como árboles jerárquicos (véase la Figura 5.37).

Formalmente, el método primero calcula una matriz de similitudes entre las neuronas  $\tilde{S}$ , cuyos elementos quedan definidos mediante  $\tilde{s}_{i,j} = f(d_{i,j})$ , donde  $d_{i,j}$  es una medida de disimilitud entre las neuronas  $\eta_i$  y  $\eta_j$ . En concreto,  $d_{i,j} = \|w_i - w_j\|$ , donde  $\|\cdot\|$  denota la distancia Euclídea. La función  $f$  es monótona decreciente para que  $\tilde{s}_{i,j}$  corresponda a una medida de similitud, donde  $f(0) = 1$  y  $f \rightarrow 0$  según aumenta su argumento de disimilitud (la distancia). En [Him00] se utiliza de forma heurística:

$$\tilde{s}_{i,j} = f(d_{i,j}) = \exp\left(-\frac{d_{i,j}^2}{\theta}\right) \quad (5.14)$$

donde  $\theta$  es un parámetro libre que controla la “fuerza de atracción” de las neuronas. El siguiente paso consiste en normalizar  $\tilde{S}$  para generar una nueva matriz  $S$ , donde  $s_{i,j} = \tilde{s}_{i,j} / \sum_{j=1}^M \tilde{s}_{i,j}$ ,  $i = 1, \dots, M$ , de tal forma que cada fila de  $S$  sume

uno. Por último, una vez definidas las coordenadas iniciales para las neuronas ( $r_i$ ), se construye un vector inicial  $X_0 = (r_1, r_2, \dots, r_M)^T$  y el proceso de contracción queda definido por:

$$X_{i+1} = SX_i, \quad i = 0, 1, \dots, r \quad (5.15)$$

o

$$X_r = S^r X_0 \quad (5.16)$$

donde  $r$  denota el número de sucesivos promediados.

El método converge a un punto que resulta ser una media ponderada de las posiciones iniciales de las neuronas. Sea  $S^* = \lim_{r \rightarrow \infty} S^r$ , el método converge a  $S^* X_0$ , donde lógicamente todas las filas de  $S^*$  son iguales y el valor en cada columna  $i$  representa el peso asociado a cada neurona  $\eta_i$  a la hora de crear la media ponderada.

Desde un punto de vista práctico el analista dispone de dos alternativas a la hora de aplicar este método. Si el objetivo consiste en obtener una rápida representación de la distribución modelada por los vectores de referencia, es conveniente reducir el número de iteraciones del algoritmo  $r$ , por lo que el parámetro  $\theta$  debe ser relativamente elevado. Por el contrario, el analista puede estar interesado también en observar la evolución de los puntos proyectados a medida que se suceden las iteraciones del algoritmo (el árbol jerárquico), por razones exploratorias, para obtener visualizaciones “suaves” del proceso dinámico de *clustering* o contracción [Him00]. En ese caso, el número de iteraciones del algoritmo  $r$  debe ser aumentado (por ejemplo, hasta varias decenas), lo cual alarga el tiempo de ejecución del algoritmo, cobrando especial importancia su complejidad computacional, véase la Sección 6.3.4. Por otro lado, cuando el número de iteraciones es elevado el analista debe evitar que las neuronas converjan a un solo punto. Esto puede ser llevado a cabo escogiendo un valor de  $\theta$  moderado, o reescalando las posiciones de las neuronas (realizando progresivos *zooms*).

La Figura 5.38 muestra un ejemplo con un MAO unidimensional de las diferentes proyecciones y árboles jerárquicos que pueden ser generados con este modelo de contracción para diferentes valores de  $\theta$ , que toma los valores 1, 0,5 y 0,07 en (a), (b) y (c), respectivamente. Para acelerar el proceso  $r$  (en (5.16)) puede ser escogido como  $r = 2^i$ , por lo que la matriz  $S$  es elevada a dos progresivamente. Los ejes de ordenadas de las imágenes (a), (b) y (c) representan el valor  $i$ . Por último, el dendograma asociado a un *clustering* aglomerativo de enlace simple es

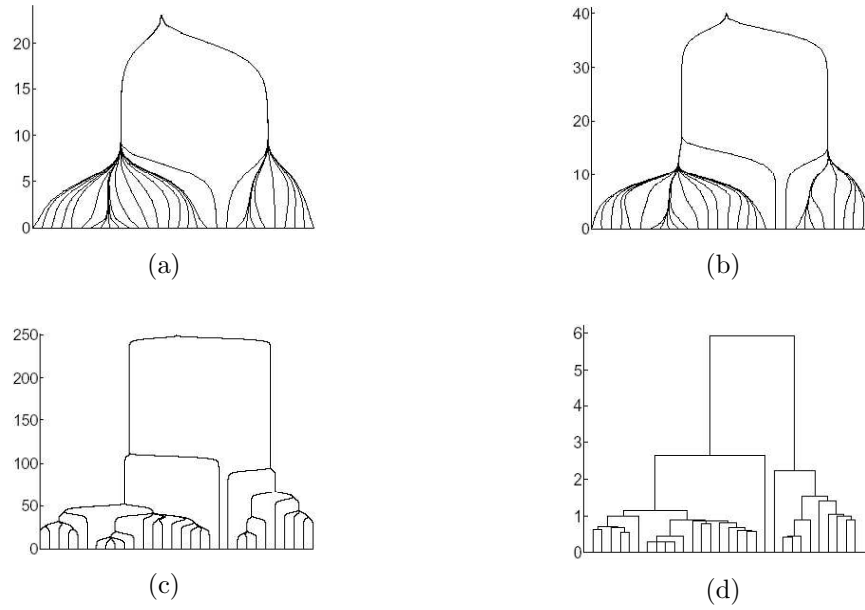


Figura 5.38: Proceso de contracción del método de Himberg para  $\theta = 1, 0.5$  y  $0.07$  en (a), (b) y (c), respectivamente, donde el eje de ordenadas representa  $\log_2 r$ . La imagen (d) ilustra el dendrograma asociado a un *clustering* aglomerativo de enlace simple, donde el eje de ordenadas representa distancias entre *clusters*. Tomado de [Him00].

ilustrado en (d), donde el eje de ordenadas muestra distancias entre *clusters*. Parece ser que a medida que  $\theta \rightarrow 0$  la proyección del método es similar a este último dendrograma, si el eje de ordenadas tiene una escala logarítmica con respecto a  $r$ , como ocurre en este caso [Him00].

La Figura 5.39 ilustra un ejemplo de la proyección que genera el método y el correspondiente encadenado con la proyección original regular del MAO, que genera una imagen coloreada de la estructura de los *clusters*. Para facilitar la comprensión del método la distribución utilizada es la formada por cuatro *clusters* disjuntos, a la que se ha adaptado un MAO  $(20 \times 20)$ -dimensional. En (a) aparece la proyección original del MAO y dos códigos de colores lineales (planos  $B = 0$  y  $B = 255 - G$  del cubo RGB) que van a ser aplicados para construir el encadenado. El resto de imágenes muestra las proyecciones obtenidas para  $\theta = 0.01$  y  $r = 1, 2$  y  $4$ , en (b), (c) y (d), respectivamente. Obsérvese cómo las neuronas asociadas a cada *cluster* son atraídas entre sí, formando progresivamente grupos compactos de puntos. Las neuronas interpolantes también aparecen claramente entre cada *cluster*. Estas proyecciones apenas varían al aumentar el número de iteraciones del algoritmo  $r$ . En las imágenes coloreadas el color que recibe cada *cluster* aparece más uniforme a medida que aumenta  $r$ , y las neuronas interpolantes aparecen

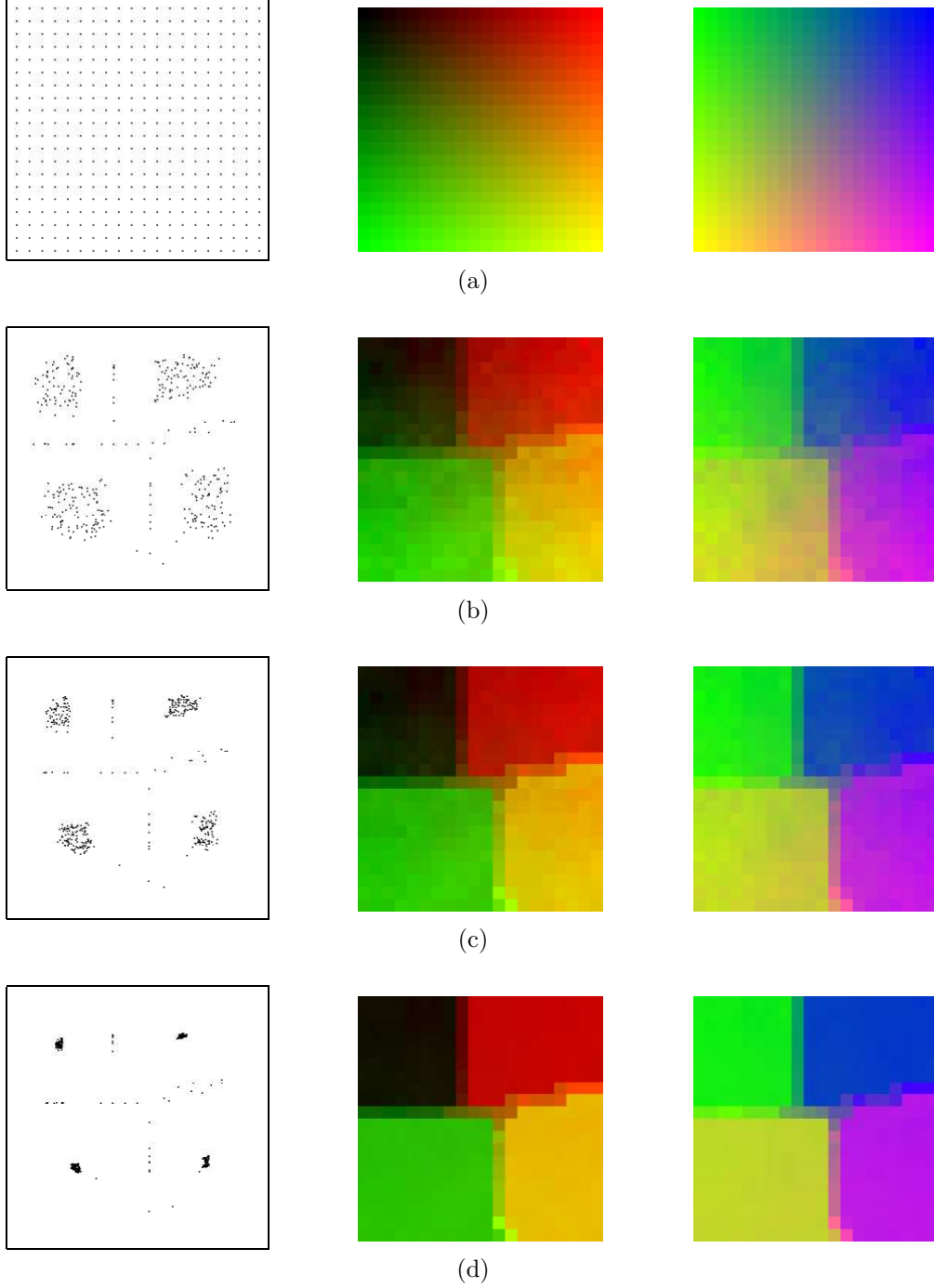


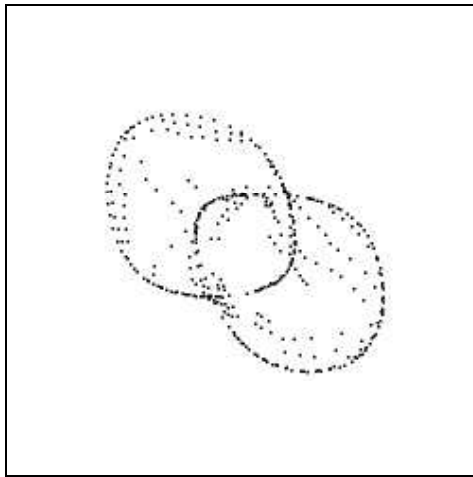
Figura 5.39: Proceso de contracción del método de Himberg y encadenado basado en dos códigos de colores lineales (planos  $B = 0$  y  $B = 255 - G$  del cubo RGB) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos. Las proyecciones mostradas utilizan  $\theta = 0,01$  y  $r = 0$  (proyección inicial del MAO), 1, 2 y 4, en (a-d), respectivamente.



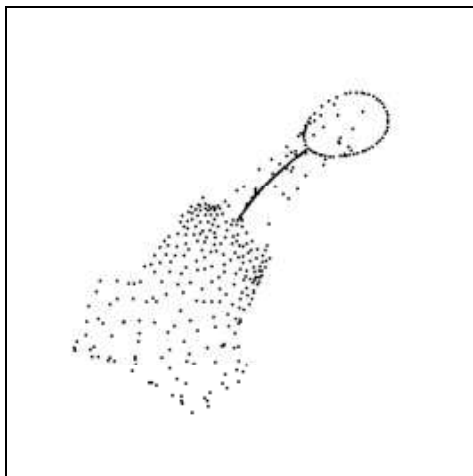
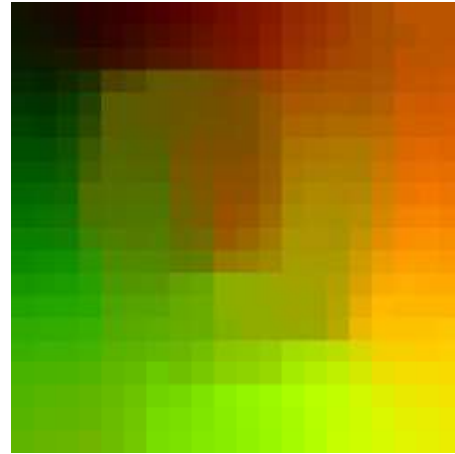
también con mayor claridad. En este caso el resultado es independiente del código de colores utilizado. No obstante, esta tesis utiliza principalmente el plano  $B = 0$  del cubo RGB. Desde un punto de vista subjetivo parece producir mejores resultados en los experimentos, donde los MAO utilizados son cuadrados.

La Figura 5.40 muestra otros dos ejemplos de las proyecciones que genera el método y sus respectivos encadenados. Los MAO son  $(20 \times 20)$ -dimensionales y están adaptados a las distribuciones formadas por dos eslabones encadenados (a) y por una circunferencia, un segmento, un cuadrado y un ortoedro (b). En (a)  $\theta = 1$  y  $r = 1$ , mientras  $\theta = 3$  y  $r = 1$  en (b). En estos casos, las proyecciones generadas permiten distinguir las clases que forman las distribuciones, y hasta revelan su forma. Por el contrario, las imágenes coloreadas son más difíciles de interpretar (sin la ayuda de la proyección del método), ya que la dimensión intrínseca de varias clases es uno, por lo que no forman *clusters* compactos. Nótese que el algoritmo de proyección fue diseñado para analizar la estructura de *clusters* mediante el encadenado. En cualquier caso, aportan información similar a las matrices de distancias, donde es posible distinguir las diferentes clases (véanse las Figuras 5.6 y 5.13).

Con respecto a la complejidad computacional del método, cuando  $\theta$  es demasiado bajo es necesario elevar a dos  $S$  tal y como se ha realizado en la Figura 5.38, lo cual implica una complejidad de  $\mathcal{O}(RM^3)$ , donde  $M$  es el número de neuronas del MAO y  $R$  corresponde al número de iteraciones, el cual puede ser elevado. El *clustering* de enlace simple está más cerca de  $\mathcal{O}(M^2)$ . De todas formas, no tiene sentido aplicar el algoritmo con un  $\theta$  demasiado bajo para obtener una versión de proceso de enlace simple. En cambio, lo interesante del proceso puede consistir en visualizar sólo las primeras iteraciones del algoritmo, por razones exploratorias, con lo que la complejidad se aproxima más a  $\mathcal{O}(RM^2)$  [Him00]. Por otro lado, si el analista sólo está interesado en analizar la proyección final de los vectores de referencia (no le interesa observar el proceso dinámico de contracción), una opción con la que es posible obtener descripciones precisas de las distribuciones consiste en escoger un valor de  $\theta$  elevado y ejecutar el algoritmo en una sola iteración, como en los ejemplos ilustrados en la Figura 5.40. Esto reduce la complejidad computacional a  $\mathcal{O}(M^2)$ . Nótese que, independientemente del número de iteraciones a ejecutar, es importante que el tamaño del MAO ( $M$ ) no sea pequeño, ya que la proyección que genera el método debe contener los suficientes puntos como para describir adecuadamente la distribución de los vectores de referencia.



(a)



(b)

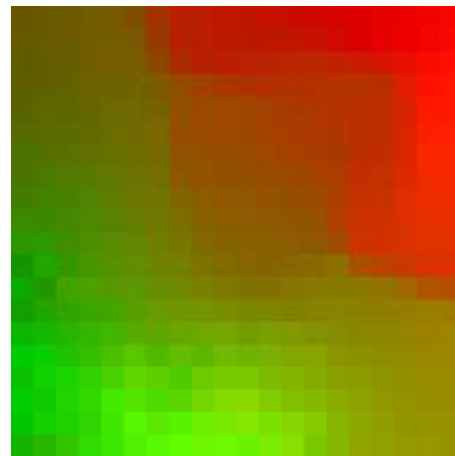


Figura 5.40: Proyección de Himberg y encadenado basado en un código de colores lineal (plano  $B = 0$  del cubo RGB) para dos MAO  $(20 \times 20)$ -dimensionales adaptados a las distribuciones formadas por dos eslabones encadenados (a) y por una circunferencia, un segmento, un cuadrado y un ortoedro (b).

En la práctica el método puede ser optimizado almacenando en memoria las distancias entre cada par de vectores de referencia del MAO. En este caso, es importante tener en cuenta detalles como el tamaño de la memoria RAM del sistema, ya que la velocidad del algoritmo puede ser reducida considerablemente si la información de las distancias no cabe en ésta (sería necesario almacenar información en un disco del sistema). Mediante esta optimización un porcentaje elevado del tiempo de ejecución puede corresponder al proceso inicial asociado a almacenar las distancias entre los vectores de referencia. De esta manera, el algoritmo puede ejecutar varias decenas de iteraciones en un tiempo razonable, sin que éste diste demasiado del tiempo de ejecución de una sola iteración (siempre que el tamaño del MAO no sea elevado). De no calcular dichas distancias el tiempo de ejecución del algoritmo resulta ser aproximadamente el producto del número de iteraciones por el tiempo necesario en calcular una de ellas, por lo que puede ser significativamente más elevado que el tiempo necesario en calcular una iteración. La Sección 6.3.4 compara la velocidad de este método con la del algoritmo de agrupación de neuronas, propuesto en la Sección 6.3.

La Figura 5.40 ilustra dos proyecciones del método para una sola iteración  $r = 1$  del algoritmo y un valor de  $\theta$  relativamente elevado, que resultan ser bastante precisas en el sentido de representar adecuadamente las clases y las relaciones entre los diferentes elementos proyectados. De hecho, es posible observar como el método de Himberg produce mejores resultados (más precisos) cuando el parámetro  $\theta$  es relativamente elevado, al menos con los conjuntos de datos utilizados en esta tesis. De esta manera, las proyecciones generadas con una sola iteración producen los resultados más precisos. Las Figuras 5.41 y 5.42 ilustran varias proyecciones finales obtenidas con diversos parámetros para un MAO  $(32 \times 32)$ -dimensional adaptado a la distribución formada por dos eslabones en-cadenados, y otro  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro, respectivamente. En la Figura 5.41 los parámetros empleados son:  $\theta = 3$ ,  $r = 1$  (a);  $\theta = 0,2$ ,  $r = 10$  (b);  $\theta = 0,1$ ,  $r = 25$  (c);  $\theta = 0,05$ ,  $r = 50$  (d);  $\theta = 0,03$ ,  $r = 100$  (e); y  $\theta = 0,01$ ,  $r = 500$  (f). En la Figura 5.42 los parámetros empleados son:  $\theta = 3$ ,  $r = 1$  (a);  $\theta = 0,3$ ,  $r = 10$  (b);  $\theta = 0,15$ ,  $r = 25$  (c);  $\theta = 0,1$ ,  $r = 50$  (d);  $\theta = 0,07$ ,  $r = 100$  (e); y  $\theta = 0,03$ ,  $r = 500$  (f). En todos los casos se ha intentado producir una proyección del mismo tamaño en el espacio de observación, es decir, las proyecciones no han sido reescaladas. Visto de otro modo, para un número determinado de

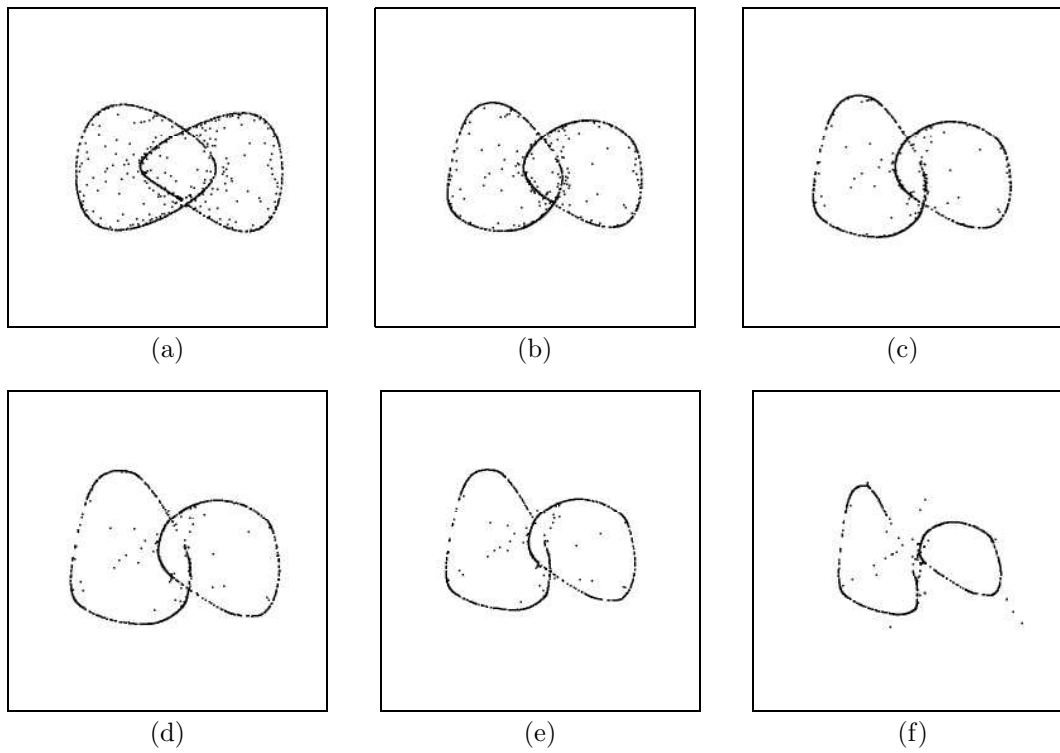


Figura 5.41: Proceso de contracción del método de Himberg para un MAO  $(32 \times 32)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados. Los parámetros utilizados son:  $\theta = 3$ ,  $r = 1$  (a);  $\theta = 0,2$ ,  $r = 10$  (b);  $\theta = 0,1$ ,  $r = 25$  (c);  $\theta = 0,05$ ,  $r = 50$  (d);  $\theta = 0,03$ ,  $r = 100$  (e);  $\theta = 0,01$ ,  $r = 500$  (f).

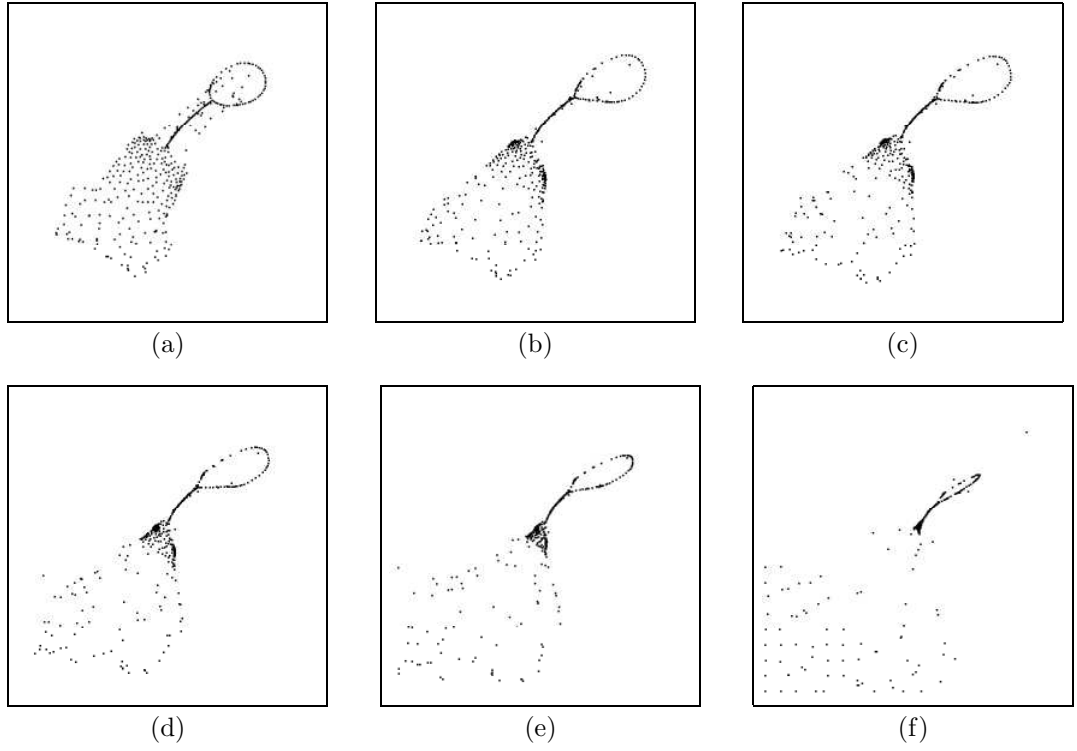


Figura 5.42: Proceso de contracción del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro. Los parámetros utilizados son:  $\theta = 3$ ,  $r = 1$  (a);  $\theta = 0,3$ ,  $r = 10$  (b);  $\theta = 0,15$ ,  $r = 25$  (c);  $\theta = 0,1$ ,  $r = 50$  (d);  $\theta = 0,07$ ,  $r = 100$  (e);  $\theta = 0,03$ ,  $r = 500$  (f).

iteraciones se ha buscado un valor de  $\theta$  apropiado para generar una proyección lo más similar posible a la primera, ejecutada en una iteración.

En los ejemplos es posible observar como la calidad de las proyecciones es degradada a medida que disminuye  $\theta$  y aumenta  $r$ . Esto tiene dos consecuencias importantes en esta tesis. En primer lugar, la proyección generada a partir de una iteración va a servir como referencia a la hora de comparar la precisión y calidad de las proyecciones de este método con respecto a las generadas con otros métodos. Por otro lado, si se desea analizar el proceso dinámico de contracción con el método de Himberg, es necesario tener en cuenta que las proyecciones pueden aparecer cada vez más distorsionadas a medida que aumenta el número de iteraciones del algoritmo. De esta manera, cabe la posibilidad de emplear otros métodos de contracción más rápidos para visualizar el proceso dinámico de contracción, aunque éstos no produzcan resultados tan precisos como el método de Himberg aplicado en una sola iteración, como el algoritmo de agrupación de neuronas (véase la Sección 6.3).

Las proyecciones obtenidas con el método de Himberg han proporcionado buenos resultados con las distribuciones de simulación propuestas en el Capítulo 4. Por su relación con el algoritmo de agrupación de neuronas, la Sección 6.3 incluye varios ejemplos de estas proyecciones y compara ambos métodos. En general, el método de Himberg produce descripciones más precisas de los datos, pero en un tiempo de ejecución mayor cuando los MAO son de tamaño elevado, y especialmente si el objetivo es analizar la evolución de las neuronas a través del proceso de contracción.

#### 5.5.2.4. Coordenadas adaptativas

Este apartado presenta el método coordenadas adaptativas [MR97], que representa una alternativa para facilitar la detección de bordes entre *clusters* en un MAO. Genera una proyección no-lineal de los vectores de referencia de un MAO, que en ocasiones resulta similar a las obtenidas con los métodos de contracción. El método está basado en los movimientos o actualizaciones de los vectores de referencia en el espacio de los datos durante el proceso secuencial de aprendizaje, los cuales se intenta imitar de alguna manera mediante desplazamientos de las neuronas en el espacio de observación. El resultado es una imagen del *clustering* de los vectores de referencia en el espacio de los datos [MR97].

El método es una extensión de la regla de aprendizaje para capturar los movimientos de los vectores de referencia en un espacio “virtual” de observación. En el paso inicial, las posiciones de los vectores proyectados  $r_i$  (denominadas coordenadas adaptativas) corresponden a las de las neuronas en la rejilla regular, como en el método anterior. Durante cada iteración del algoritmo secuencial de entrenamiento, las distancias entre los vectores de referencia  $w_i$  y el nuevo dato  $x$  escogido al azar que es presentado al algoritmo son calculadas y almacenadas en una tabla  $D_i(t)$ . Después de la adaptación de los vectores de referencia, donde se identifica una neurona ganadora  $\eta_c$ , se calcula una tabla de distancias similar  $D_i(t+1)$ . El siguiente paso consiste en calcular el cambio relativo de distancias para cada neurona  $\eta_i$  del MAO:

$$D'_i(t+1) = \frac{D_i(t) - D_i(t+1)}{D_i(t)} \quad (5.17)$$

Finalmente, las coordenadas adaptativas de todas las neuronas excepto la ganadora son actualizadas en dirección hacia la ganadora por la fracción dada en  $D'_i(t+1)$ , quedando definido el proceso mediante la siguiente expresión:

$$r_i(t+1) = r_i(t) + D'_i(t+1)(r_c(t) - r_i(t)) \quad (5.18)$$

En la práctica, según avanza el entrenamiento, los vectores proyectados son atraídos entre sí, lo cual genera visualizaciones en las que es posible apreciar *clusters* y especialmente bordes entre ellos. Sin embargo, si la tasa de aprendizaje del algoritmo es elevada los puntos pueden converger rápidamente a una única localización, o si es baja puede que no sean desplazados lo suficiente. Por este motivo, en esta tesis se ha introducido un parámetro nuevo  $\alpha$  en (5.18), para mitigar o reforzar la fuerza de atracción de las neuronas de cara a la visualización, quedando finalmente la siguiente regla para calcular la proyección:

$$r_i(t+1) = r_i(t) + \alpha D'_i(t+1)(r_c(t) - r_i(t)) \quad (5.19)$$

La Figura 5.43 ilustra un ejemplo de la proyección del método y la correspondiente imagen tras realizar el encadenado. La distribución corresponde a la formada por cuatro *clusters* disjuntos, a la que se ha adaptado un MAO ( $20 \times 20$ )-dimensional. Nótese la similitud con respecto a las imágenes ilustradas en la Figura 5.39 (a pesar de que los datos y MAO utilizados son diferentes). En este

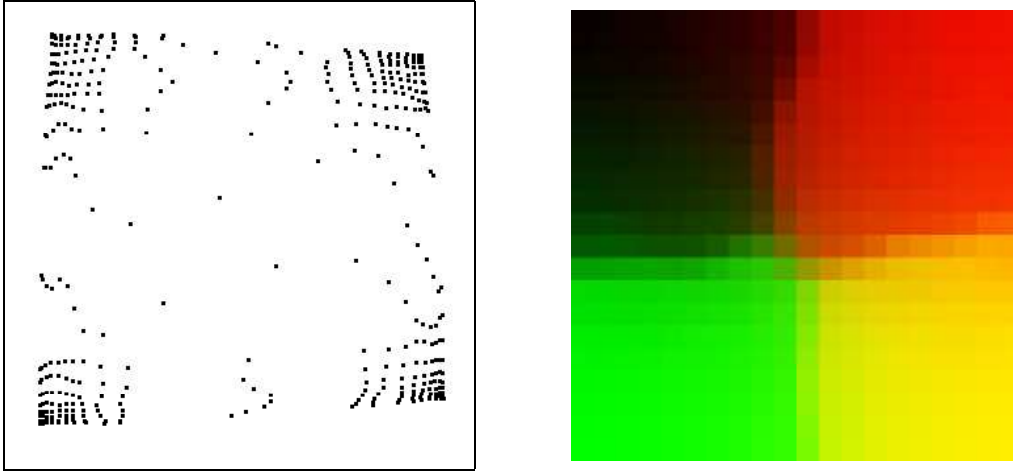


Figura 5.43: Proyección final del método coordenadas adaptativas y encadenado basado en un código de colores lineal para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos.

caso las neuronas también son atraídas para formar *clusters* compactos.

El método presenta dos principales desventajas. En primer lugar, los métodos de contracción generalmente producen descripciones más precisas de los conjuntos de datos utilizados en el entrenamiento de los MAO. La Figura 5.44 ilustra las proyecciones generadas por el método y sus respectivos encadenados para dos MAO  $(20 \times 20)$ -dimensionales adaptados a las distribuciones formadas por dos eslabones encadenados (a) y por una circunferencia, un segmento, un cuadrado y un ortoedro (b) (véanse las mismas proyecciones generadas por el modelo de contracción en la Figura 5.40). En (a) es posible detectar los dos anillos, pero no con la misma claridad que con los métodos de contracción. En (b) resulta muy difícil distinguir las cuatro clases que componen la distribución de datos. La falta de precisión de este método también queda reflejada en las imágenes coloreadas, resultado de aplicar el encadenado.

La otra desventaja reside en la necesidad de generar la visualización a la vez que el MAO es entrenado, lo cual puede repercutir negativamente en su velocidad y en las necesidades de almacenamiento. Nótese que generalmente se entrenan muchos MAO con el objetivo de elegir uno “óptimo”. Por un lado, la complejidad computacional del algoritmo es  $\mathcal{O}(MT)$ , donde  $M$  es el número de neuronas del MAO y  $T$  es el número de iteraciones del proceso de aprendizaje, que generalmente suelen ser elevados. Por otro, el analista está obligado a almacenar las proyecciones de cada iteración o la información suficiente como para repetir un



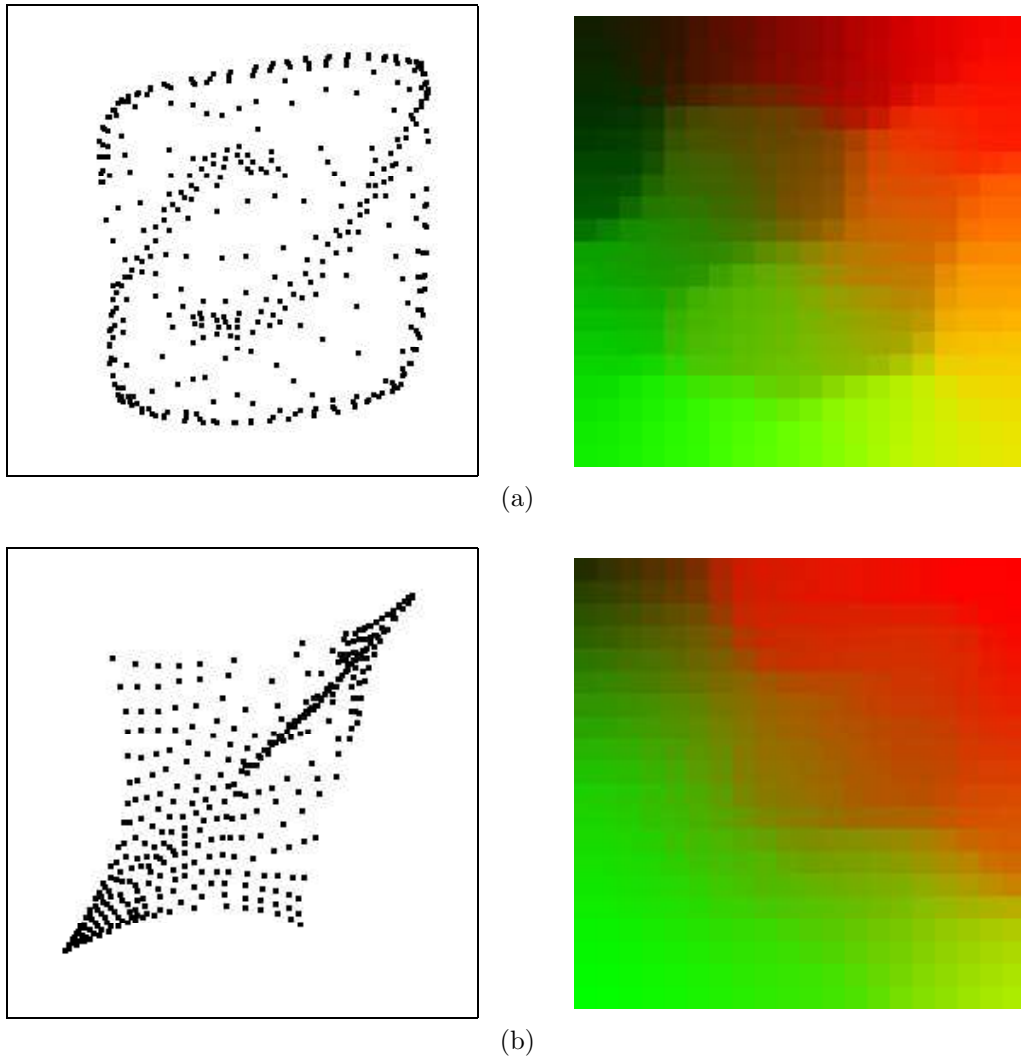


Figura 5.44: Proyección final del método coordenadas adaptativas y encadenado basado en un código de colores lineal para dos MAO  $(20 \times 20)$ -dimensionales adaptados a las distribuciones formadas por dos eslabones encadenados (a) y por una circunferencia, un segmento, un cuadrado y un ortoedro (b).

entrenamiento idéntico al llevado a cabo si se desea analizar el proceso repetidas veces. La repetición del proceso es importante por varias razones:

- El parámetro  $\alpha$  generalmente debe ser modificado para conseguir una visualización efectiva.
- El analista no puede centrar su atención en todas las regiones del espacio de observación al mismo tiempo.
- El análisis del proceso dinámico de contracción es complejo. Resulta mucho más difícil analizar multitud de imágenes en una secuencia que una sola estática. Los árboles tridimensionales (véase la Figura 5.37 (b)) pueden representar una solución a éste problema, pero su análisis e implementación también son muy complejos.

Por estas razones, este método no ha sido considerado en la Sección 6.3.4, donde se comparan los tiempos de ejecución del algoritmo de agrupación de neuronas y del método de Himberg (ambos métodos de contracción de neuronas).

## Capítulo 6

# NUEVOS MÉTODOS DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS

### 6.1. Método de Semejanza de Triángulos

La idea de proyectar los datos de la distribución con la que se ha entrenado un MAO sobre un espacio de observación continuo no ha sido explotada en la literatura de visualización de MAO. Las estrategias de interpolación propuestas (véanse las Secciones 5.2.3 y 5.4.4) generalmente no producen descripciones adecuadas de los conjuntos de datos, o aportan la misma información que otros métodos de visualización.

Este apartado presenta un nuevo método de interpolación más sofisticado y robusto que los anteriores, el método de semejanza de triángulos (MST) [RDG<sup>+</sup>02, RG03], capaz de solventar varias deficiencias relacionadas con éstos al poder considerar más de tres neuronas a la hora de calcular la proyección de cada dato. El método está basado en la correspondencia discreta entre los vectores de referencia de las neuronas y las posiciones de éstas en el mapa, creando una proyección no-lineal desde el espacio de los datos al espacio de observación del MAO. Un patrón determinado es proyectado en un intento por preservar la semejanza geométrica entre varios triángulos formados por dicho patrón y dos vectores de referencia en el espacio de los datos, y por un punto candidato y dos neuronas en el espacio de observación. El método encuentra la proyección de un dato minimizando una función de coste que mide distancias o errores entre varios triángulos. El MST

está relacionado con métodos de EMD [Tor52, Sam69] y con el método de triangulación [LSB77], y su objetivo es preservar relaciones de proporcionalidad entre distancias en función de un conjunto de puntos fijos. En esta tesis los puntos fijos son las neuronas en la rejilla regular del MAO y sus vectores de referencia en el espacio de entrada, aunque el método puede ser aplicado igualmente con otras proyecciones que generen diferentes localizaciones de un conjunto de centroides o vectores de referencia en un espacio observable. Este método resulta atractivo desde el punto de vista del AED ya que proyecta todos los datos de manera no-lineal. También resulta una técnica de visualización adecuada cuando el tamaño del MAO es pequeño (donde suelen tener problemas la mayoría de métodos de visualización), es robusto a la adaptación del MAO a la distribución en cuanto al error de cuantización y puede describir adecuadamente ciertos tipos de distribuciones difíciles de visualizar con la mayoría de métodos de visualización.

### 6.1.1. Descripción del problema

Los métodos de EMD (véase la Sección 2.5.5.1) primero proyectan todos los datos de una distribución de dimensión elevada a un espacio observable (de manera aleatoria, según una proyección lineal, etc.) y posteriormente actualizan todas las posiciones de los datos proyectados minimizando una función de coste, donde persiguen preservar todas las distancias originales (absolutas) entre los datos de la distribución. Este apartado describe una nueva estrategia para proyectar los datos. El objetivo consiste en proyectar primero un conjunto pequeño de elementos al espacio observable para posteriormente proyectar la distribución, dato a dato, en función únicamente del primer conjunto proyectado, que sirve como referencia. De esta manera, se desea preservar las relaciones de proporcionalidad entre las distancias desde cada nuevo dato a los elementos del conjunto de referencia. En esta tesis el conjunto de referencia queda definido mediante la correspondencia discreta entre las neuronas y los vectores de referencia de un MAO adaptado a la distribución de datos. El método de interpolación propuesto puede ser aplicado también con otros conjuntos de referencia, por ejemplo, una serie de centroides calculados a través de un algoritmo de CV o un subconjunto pequeño de la distribución de datos, los cuales pueden ser proyectados a un espacio observable mediante alguna técnica de proyección (lineal, basada en el EMD, etc.). La idea de proyectar los datos secuencialmente también es utilizada en el método de triangulación [LSB77], el cual preserva las distancias absolutas a dos elementos

previamente proyectados.

Una vez entrenado un MAO básico con una distribución de patrones de entrada  $\Delta = \{x_i\}$ ,  $i = 1, \dots, N$ , donde  $x_i \in \mathbb{R}^n$ , se pretende proyectar cada elemento  $x$  a un espacio observable continuo  $\mathbb{R}^m$ , siguiendo la filosofía de los métodos de interpolación. Es decir, se desea construir una función  $f$  (véase (5.6)) no-lineal, teniendo como referencia la correspondencia discreta  $f_d$  (véase (3.1)) entre el conjunto  $\Omega = \{w_i\}$ ,  $i = 1, \dots, M$  de vectores de referencia del MAO, y el conjunto  $\Phi = \{r_i\}$ ,  $i = 1, \dots, M$  de las posiciones de las neuronas.

Este problema de interpolación no es trivial ya que no es posible realizar ninguna suposición sobre  $f$ . Sin embargo, además de la condición de que el MAO preserve la topología adecuadamente, destacan las siguientes propiedades, *a priori* deseables, que debe seguir  $f$  para crear el algoritmo:

- Continuidad. Vectores de entrada muy similares deben proyectarse a zonas del mapa (espacio observable) muy próximas.
- Ajuste. Los propios vectores de referencia deben ser proyectados a las posiciones de sus neuronas asociadas en el mapa:

$$f(w_i) = f_d(w_i) = r_i \quad \forall w_i \in \Omega \quad (6.1)$$

En la práctica esta propiedad es opcional ya que el analista puede desear generar una proyección libre de la configuración regular de las neuronas en el espacio de observación.

- Robustez. El método debe ser capaz de encontrar una proyección adecuada incluso cuando el MAO no se haya ajustado perfectamente a la distribución con un error de cuantización bajo.
- Velocidad. El algoritmo que implementa la función  $f$  debe ser capaz de proyectar cada dato en poco tiempo.

La propiedad más importante a la hora de proyectar adecuadamente una distribución de datos es la continuidad, ya que toda la información que aporta la proyección está basada en las distancias entre los elementos. El analista sólo será capaz de interpretar la estructura de los datos y relacionar los diferentes elementos aparecidos en el mapa si la proyección es continua o si las discontinuidades son pequeñas. Por ejemplo, si la proyección genera discontinuidades apreciables

no será posible observar adecuadamente un *cluster* si diferentes subconjuntos de éste aparecen en diferentes lugares separados en el mapa.

Un MAO puede ser visto como un conjunto de vectores de referencia distribuidos y ordenados en una rejilla regular que han sido adaptados a una distribución de datos. La propiedad de ajuste debe ser satisfecha para que el método proyecte los datos siguiendo la configuración de los vectores de referencia en el mapa. Esto puede ser útil de cara a analizar la estructura local de los datos o comparar el método de interpolación con otro método de visualización de MAO, donde la posición de los vectores de referencia o sus neuronas asociadas no varíe (como suele ocurrir en la mayoría de los métodos).

Por otro lado, si el objetivo es analizar la estructura global de la distribución la propiedad de ajuste puede ser ignorada para dar más libertad a la proyección para desvincularse de la configuración fija de las neuronas y sus vectores de referencia. En este caso puede no ser estrictamente necesario que los vectores de referencia del MAO se adapten a la distribución con un error de cuantización bajo, siempre que éstos estén ordenados para poder servir como referencia a la proyección. De esta manera, la robustez del método puede ser importante cuando la dimensión intrínseca de los datos es mayor que la dimensión del MAO, ya que se podrá trabajar con un MAO que optimice la preservación la topología aunque el error de cuantización medio sea relativamente elevado.

Por último, dado el tamaño de las actuales bases de datos, un análisis exhaustivo puede requerir proyectar millones de datos. En estos casos la velocidad del método de interpolación resulta ser un factor decisivo a la hora de determinar su utilidad.

Los métodos de interpolación descritos en la literatura (véanse las Secciones 5.2.3 y 5.4.4) son rápidos y satisfacen la propiedad de ajuste. Sin embargo, no proporcionan una proyección continua ni robusta, por lo que generalmente no pueden aportar descripciones adecuadas de las distribuciones. El siguiente apartado describe el método de interpolación propuesto cuya principal ventaja reside en que puede generar una proyección continua y robusta.

### 6.1.2. Solución propuesta

Dado un dato de entrada  $x \in \Delta$  a proyectar, se selecciona un conjunto de pares de neuronas  $\Psi(x)$  cuyos vectores de referencia van a formar triángulos con  $x$  en el espacio de los datos. Posteriormente se pretende proyectar el dato a una

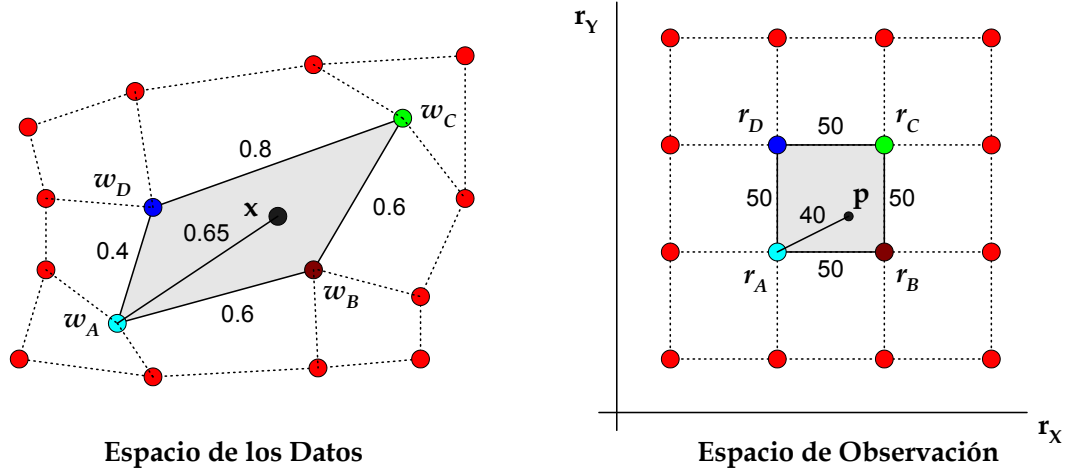


Figura 6.1: Necesidad de utilizar distancias relativas. Los vectores de referencia  $w$  y las posiciones de las neuronas  $r$  están definidas en escalas diferentes por lo que el método no puede preservar distancias absolutas a la hora de calcular la proyección de cada dato. En la figura las distancias son aproximadas.

posición en el mapa  $p$ , que a su vez va a formar un triángulo con las posiciones de los pares de neuronas de  $\Psi(x)$  en el espacio observable. La idea fundamental del método de interpolación consiste en preservar lo mejor posible las relaciones de proporcionalidad o cocientes entre las longitudes de las aristas de los triángulos formados, para que éstos sean lo más parecidos posible. Desde otro punto de vista, la idea puede ser vista como un intento de preservar la semejanza geométrica entre dichos triángulos, con lo que también puede considerarse la preservación de los ángulos de los triángulos formados.

A diferencia de los métodos de EMD o el método de triangulación, los cuales preservan las distancias originales absolutas entre los elementos, en el MST el objetivo es la preservación de distancias relativas, es decir, de las relaciones de proporcionalidad entre las distancias. No es posible utilizar distancias absolutas ya que las posiciones de las neuronas (puntos fijos del conjunto de referencia a la hora de calcular la proyección) pueden estar definidas en una escala diferente a la de los datos y vectores de referencia. La Figura 6.1 ilustra este problema, donde las distancias entre los vectores de referencia son muy distintas a las distancias entre las neuronas, las cuales pueden estar basadas, por ejemplo, en un número determinado de píxeles.

Para cada par de neuronas  $(\eta_i, \eta_j)_{i \neq j} \in \Psi(x)$  se obtiene la situación descrita en la Figura 6.2. Un patrón  $x$  deberá ser proyectado a un punto  $p$  en el mapa de

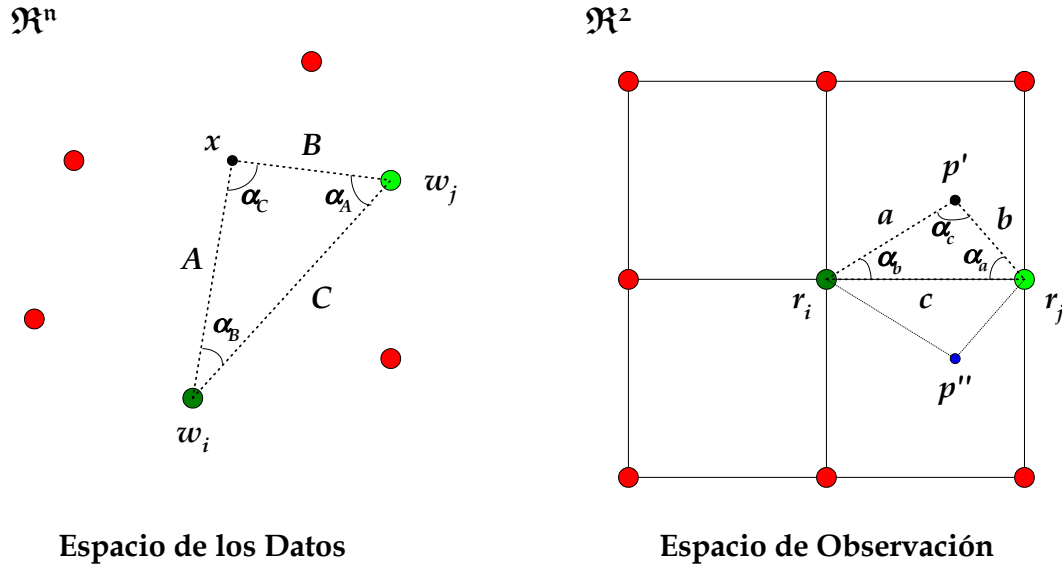


Figura 6.2: Idea principal del MST. Dado un vector de entrada  $x$  y un par de vectores de referencia  $w_i$  y  $w_j$ , se busca un punto  $p$  ( $p'$  o  $p''$ ) en el mapa de tal forma que el triángulo formado por los puntos proyectados  $(p, r_i, r_j)$  sea semejante al triángulo formado por  $(x, w_i, w_j)$ .

manera que los triángulos  $(x, w_i, w_j)$  y  $(p, r_i, r_j)$  resulten semejantes. Esto ocurre cuando:

$$\frac{A}{a} = \frac{B}{b} = \frac{C}{c}$$

donde

$$\begin{aligned} A &= \|x - w_i\|_n & B &= \|x - w_j\|_n & C &= \|w_i - w_j\|_n \\ a &= \|p - r_i\|_m & b &= \|p - r_j\|_m & c &= \|r_i - r_j\|_m \end{aligned}$$

o si

$$\alpha_A = \alpha_a, \quad \alpha_B = \alpha_b \quad \text{y} \quad \alpha_C = \alpha_c$$

Aplicando el teorema del coseno es posible calcular el valor de cada ángulo de los triángulos. Por ejemplo:

$$\alpha_A = \arccos \left( \frac{B^2 + C^2 - A^2}{2BC} \right)$$

Nótese que si  $\Psi(x)$  contiene un solo par de neuronas en MAO bidimensionales aparecen dos posibles proyecciones  $p'$  y  $p''$  en el mapa que preservan perfectamente la semejanza del triángulo original. Las dos proyecciones corresponden al lugar geométrico donde se cortan las tres circunferencias (salvo en casos degenerados) definidas por  $A/a = B/b$ ,  $A/a = C/c$  y  $B/b = C/c$ , véase la Figura 6.3, o al



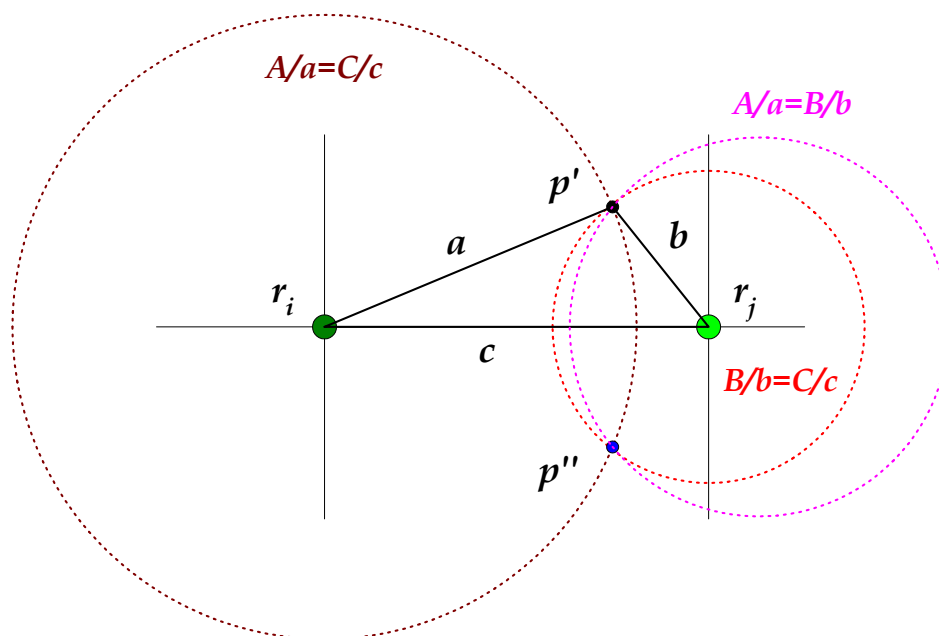


Figura 6.3: Cálculo de las proyecciones en función de las relaciones entre las longitudes de las aristas de los triángulos.

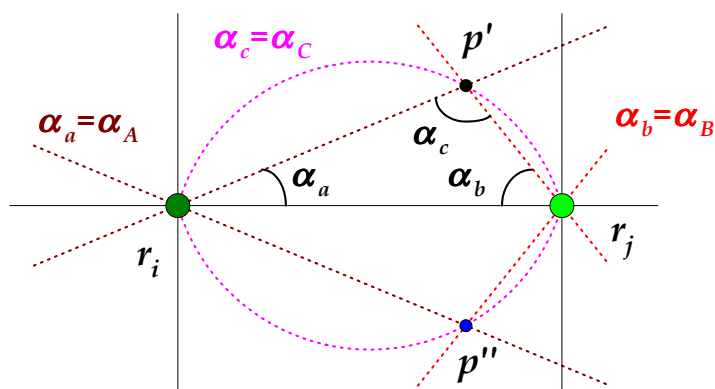


Figura 6.4: Cálculo de las proyecciones en función de los ángulos de los triángulos.

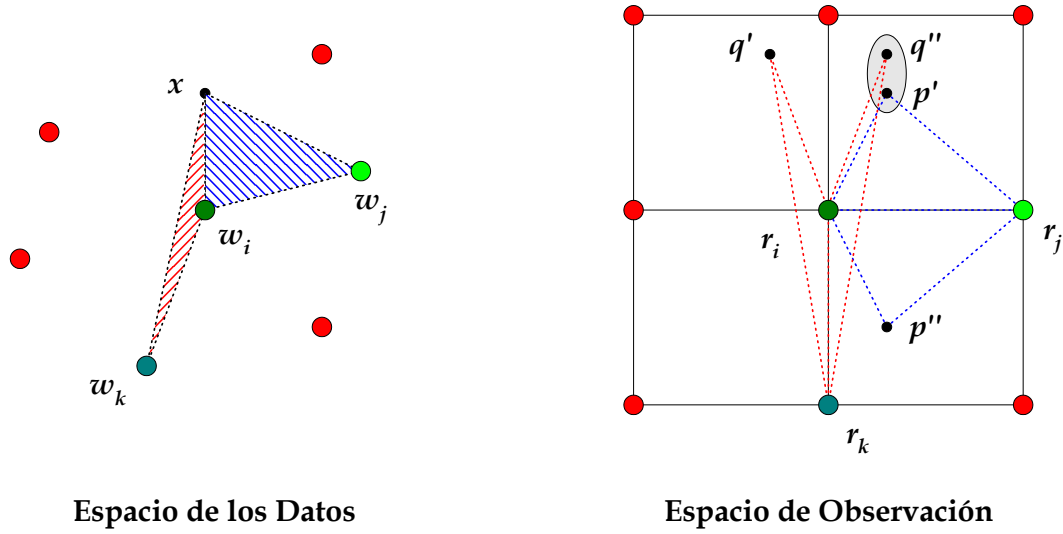


Figura 6.5: Incorporación de un segundo par de neuronas a  $\Psi(x)$ . Dos proyecciones adicionales  $q'$  y  $q''$  son generadas, que pueden servir para determinar un lugar geométrico (señalado en gris) donde la semejanza entre los triángulos sea preservada “lo mejor posible”. Por otro lado, parece más razonable intentar preservar la semejanza del triángulo  $(x, w_i, w_j)$ , donde  $p = p'$ , antes que la del  $(x, w_i, w_k)$ , ya que el segmento  $\overline{w_i w_j}$  se encuentra “más cerca” de  $x$  que el segmento  $\overline{w_i w_k}$ .

lugar geométrico donde se cortan las rectas definidas por  $\alpha_A = \alpha_a$  y  $\alpha_B = \alpha_b$ , y los arcos capaces definidos por  $\alpha_C = \alpha_c$ , véase la Figura 6.4.

Esta ambigüedad debe ser resuelta incorporando al menos otro par de neuronas a  $\Psi(x)$  para poder aportar más información sobre la ubicación final de  $p$ . Sin embargo, al considerar un nuevo par de neuronas aparecen dos nuevas proyecciones  $q'$  y  $q''$ , que en la mayoría de los casos no van a coincidir con  $p'$  o  $p''$ . Por esta razón no va a ser posible determinar con exactitud un punto  $p$  de tal forma que la semejanza geométrica entre los dos pares de triángulos sea preservada perfectamente. De todas formas, las nuevas proyecciones pueden aportar información ya que intuitivamente es posible escoger un punto  $p$  que preserve “lo mejor posible” las semejanzas entre los triángulos. Esta última idea es ilustrada en la Figura 6.5, donde parece razonable proyectar  $x$  a un punto cercano a  $p'$  y  $q''$ , ya que en esa región los triángulos formados en el mapa van a ser similares (aunque no estrictamente semejantes) a los originales en el espacio de los datos. Por otro lado, parece tener más sentido proyectar  $x$  sobre  $p'$  antes que sobre  $q''$  ya que el segmento  $\overline{w_i w_j}$  se encuentra “más cerca” de  $x$  que el  $\overline{w_i w_k}$  (definiendo un criterio para determinar la distancia desde un punto a un segmento, por ejemplo, la suma de distancias desde  $x$  a los dos vectores de referencia que componen el

segmento). De esta manera, el triángulo  $(x, w_i, w_j)$  sería preservado mejor que el  $(x, w_i, w_k)$ .

A medida que la cardinalidad de  $\Psi(x)$  aumenta y más pares de neuronas son considerados, es evidente que lo normal es no poder encontrar un punto  $p$  tal que se preserve la semejanza entre todos los triángulos. Sin embargo, parece razonable la existencia de una zona del mapa donde la mayoría de triángulos proyectados se asemejen a los originales (en cuanto a las relaciones entre las longitudes de sus aristas o a sus ángulos) que pueda ser buscada mediante un método de optimización. Por tanto, surge la necesidad de crear un criterio o error que aporte una medida de disimilitud o distancia entre triángulos basada en las diferencias entre sus ángulos o diferencias entre las relaciones de proporcionalidad entre las longitudes de sus aristas. Al buscar una situación en la que  $A/a = B/b$ ,  $A/a = C/c$  y  $B/b = C/c$ , o donde  $\alpha_A = \alpha_a$  y  $\alpha_B = \alpha_b$  (véase la Figura 6.2), se proponen las siguientes medidas de error o disimilitud entre dos triángulos:

$$\xi_1 = |Ab - aB| + |Ac - aC| + |Bc - bC| \quad (6.2)$$

$$\xi_2 = \frac{|Ab - aB| + |Ac - aC| + |Bc - bC|}{(A + B + C) \cdot (a + b + c)} \quad (6.3)$$

$$\xi_3 = \sqrt{(\alpha_A - \alpha_a)^2 + (\alpha_B - \alpha_b)^2 + (\alpha_C - \alpha_c)^2} \quad (6.4)$$

Estas medidas son igual a cero si y sólo si los triángulos son geoméricamente semejantes y su elección resulta ser un factor importante de cara a visualizar adecuadamente distribuciones en el espacio de observación de un MAO. El error  $\xi_1$  no es invariante ante el tamaño de los triángulos, sin embargo, ha generado los mejores resultados (véase la Sección 6.1.3). Una de las líneas futuras de esta tesis es investigar nuevas medidas similares (por ejemplo, pueden estar basadas en los lugares geoméricos descritos en las Figuras 6.3 y 6.4).

Una vez definida la disimilitud entre dos triángulos es posible buscar un punto  $p$  en el plano que minimice dichos errores entre cada triángulo formado por  $x$  y los vectores de referencia  $(w_i, w_j)$  de cada par de neuronas de  $\Psi(x)$ , y sus respectivos triángulos en el mapa formados por  $p$ ,  $r_i$  y  $r_j$ . Para ello es necesario combinar cada error individual de cada triángulo en una función de coste total que considere todos los triángulos formados a partir de los pares de neuronas de  $\Psi(x)$ . En esta tesis dicha función de coste es una suma ponderada de los errores entre cada

pareja de triángulos:

$$\varepsilon_x(p) = \sum_{(\eta_i, \eta_j) \in \Psi(x)} \xi(x, w_i, w_j, p, r_i, r_j) \varphi(x, w_i, w_j) \quad (6.5)$$

donde  $\xi$  es una de las medidas de disimilitud entre dos triángulos. La función  $\varphi$  sirve para ponderar la importancia relativa de preservar la semejanza de un determinado triángulo. Por ejemplo, puede ser preferible preservar los triángulos formados con vectores de referencia próximos a  $x$ . En varios experimentos presentados en la Sección 6.1.3,  $\varphi$  es una función decreciente de la suma de las distancias desde  $x$  a  $w_i$  y  $w_j$ :

$$\varphi(x, w_i, w_j) = e^{-K(A+B)} \quad (6.6)$$

donde  $A = \|x - w_i\|_n$ ,  $B = \|x - w_j\|_n$  y  $K = \ln(10)/Diametro$ , siendo *Diametro* la máxima distancia entre todos los datos de la distribución de entrada.

Por otro lado, la función  $\varphi$  puede ser escogida para construir una proyección continua si las neuronas de los pares en  $\Psi(x)$  son elegidas en función de una distancia máxima  $D$  desde sus vectores de referencia a  $x$  y se consideran todos los pares formados por todas las neuronas en  $\Psi(x)$  (como se verá a continuación). En este caso,  $\varphi$  debe ser una función continua, decreciente e igual a cero cuando  $\|x - w_i\| \geq D$  o  $\|x - w_j\| \geq D$ . Por ejemplo:

$$\varphi(x, w_i, w_j) = \begin{cases} \frac{(D-A)(D-B)}{D^2} & A < D \text{ y } B < D \\ 0 & \text{en otro caso} \end{cases} \quad (6.7)$$

donde  $A = \|x - w_i\|$ ,  $B = \|x - w_j\|$  y  $D$  es escogida por el analista. En otras palabras,  $\varphi$  debe asegurar que el conjunto de pares  $\Psi(x)$  no varíe ante pequeñas variaciones de  $x$ . En caso de introducir nuevos pares debe “ignorarlos” a la hora de calcular  $\varepsilon_x(p)$ . Esta idea es ilustrada en la Figura 6.6. En (a), a pesar de que una pequeña variación del vector de entrada introduce una nueva neurona y tres pares más en  $\Psi(x)$ , éstos son ignorados con la función de ponderación  $\varphi$ . La función descrita en (6.7) es ilustrada en (b) para  $D = 30$ .

Finalmente el MST resulta ser un problema de optimización, donde la proyección  $p$  de un dato  $x$  es calculada minimizando  $\varepsilon_x(p)$ :

$$f(x) = p \text{ / } \varepsilon_x(p) < \varepsilon_x(p_i) \quad \forall i \quad (6.8)$$

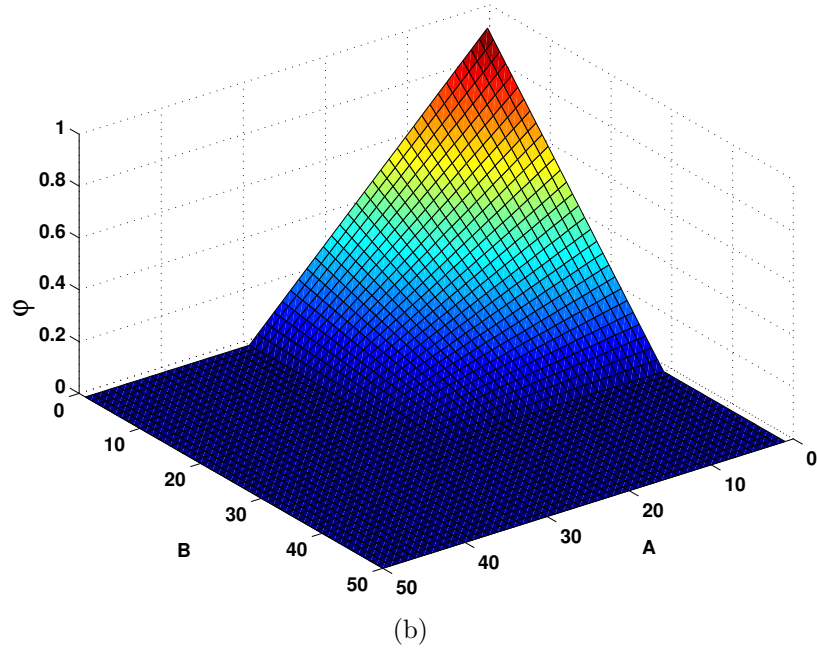
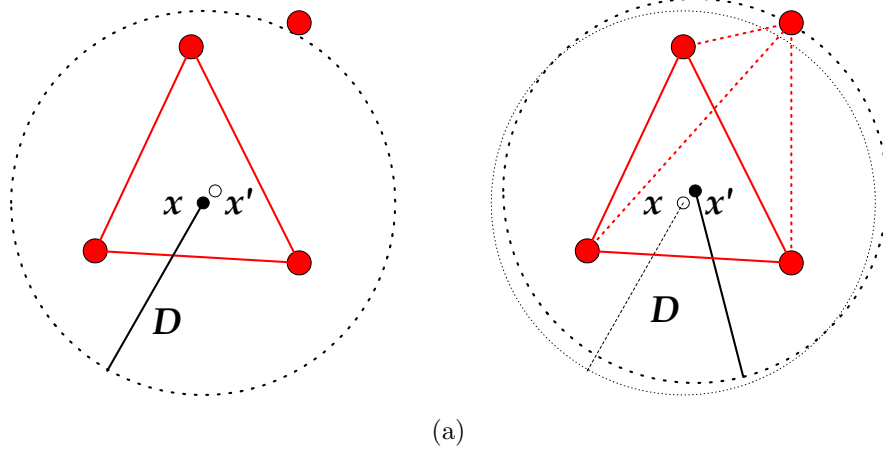


Figura 6.6: Construcción de una proyección continua mediante la función  $\varphi$ . Las aristas uniendo vectores de referencia indican pares de neuronas en  $\Psi(x)$ . En (a) un pequeño desplazamiento de  $x$  puede introducir nuevos pares que deben ser ignorados por la función de ponderación  $\varphi$ . La función descrita en (6.7) es ilustrada en (b) para  $D = 30$ .

En los experimentos se ha utilizado una técnica sencilla (y lenta) de descenso de gradiente donde el punto inicial es la posición en el mapa de la neurona ganadora  $r_c$  asociada al dato  $x$ . En algunos casos puede ser necesario considerar varios puntos iniciales para alcanzar un mínimo global de la función  $\varepsilon_x(p)$ , especialmente si se desea construir una proyección continua. En concreto, dado un punto o pixel inicial  $p$  se busca un pixel vecino tal que minimice  $\varepsilon_x(p)$ . Este proceso es repetido hasta hallar un mínimo local. Hay que destacar que el objetivo principal del estudio es evaluar la capacidad del método para proyectar datos adecuadamente, por lo que no se ha investigado la implementación de un algoritmo más rápido para calcular la optimización. De esta manera, los tiempos de ejecución descritos en el apartado de resultados experimentales (Sección 6.1.3) son relativos y pueden ser notablemente reducidos.

La Figura 6.7 muestra gráficamente la idea principal del MST. Se supone que un MAO  $(2 \times 2)$ -dimensional se ha adaptado a la distribución bidimensional formada por cuatro *clusters* de igual tamaño y diferente forma. Para un dato de entrada  $x$  se pretende hallar su proyección  $p$  en el espacio de observación del MAO, definido en una escala diferente, minimizando una determinada función de error entre los triángulos formados en ambos espacios (a). En el ejemplo  $\Psi(x)$  contiene todos los pares (seis) de neuronas posibles ( $\Psi(x) = \{(\eta_1, \eta_2), (\eta_1, \eta_3), (\eta_1, \eta_4), (\eta_2, \eta_3), (\eta_2, \eta_4), (\eta_3, \eta_4)\}$ ) y la función de ponderación es ignorada ( $\varphi = 1$ ) para construir la superficie de error  $\varepsilon_x(p)$ . Las imágenes (b), (c) y (d) muestran  $\varepsilon_x(p)$  para las medidas de disimilitud entre triángulos  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ , respectivamente. Como cabía esperar, la proyección de  $x = (100, 150)$  corresponde al mínimo de  $\varepsilon_x(p)$  en el punto  $p = (20, 30)$ .

Otra cuestión crucial es determinar el conjunto de pares de neuronas  $\Psi(x)$  (sobre todo su cardinalidad) con el que se quiere preservar la semejanza entre los triángulos, ya que éste determina las propiedades de la proyección  $f$  (continuidad, ajuste, robustez y velocidad) e influye notablemente en la calidad de las descripciones de las distribuciones de datos. Lo más razonable es escoger pares de neuronas cuyos vectores de referencia estén próximos a  $x$ . Dada la propiedad de preservación de la topología de los MAO eso es prácticamente equivalente a escoger pares de neuronas que se hallen cerca de la neurona ganadora en el mapa. En los experimentos los pares de neuronas han sido seleccionados de un conjunto previo de neuronas  $\Upsilon(x)$ , principalmente mediante las dos siguientes estrategias:

- Considerar las neuronas cuyos vectores de referencia se encuentran dentro

## 6.1. MÉTODO DE SEMEJANZA DE TRIÁNGULOS

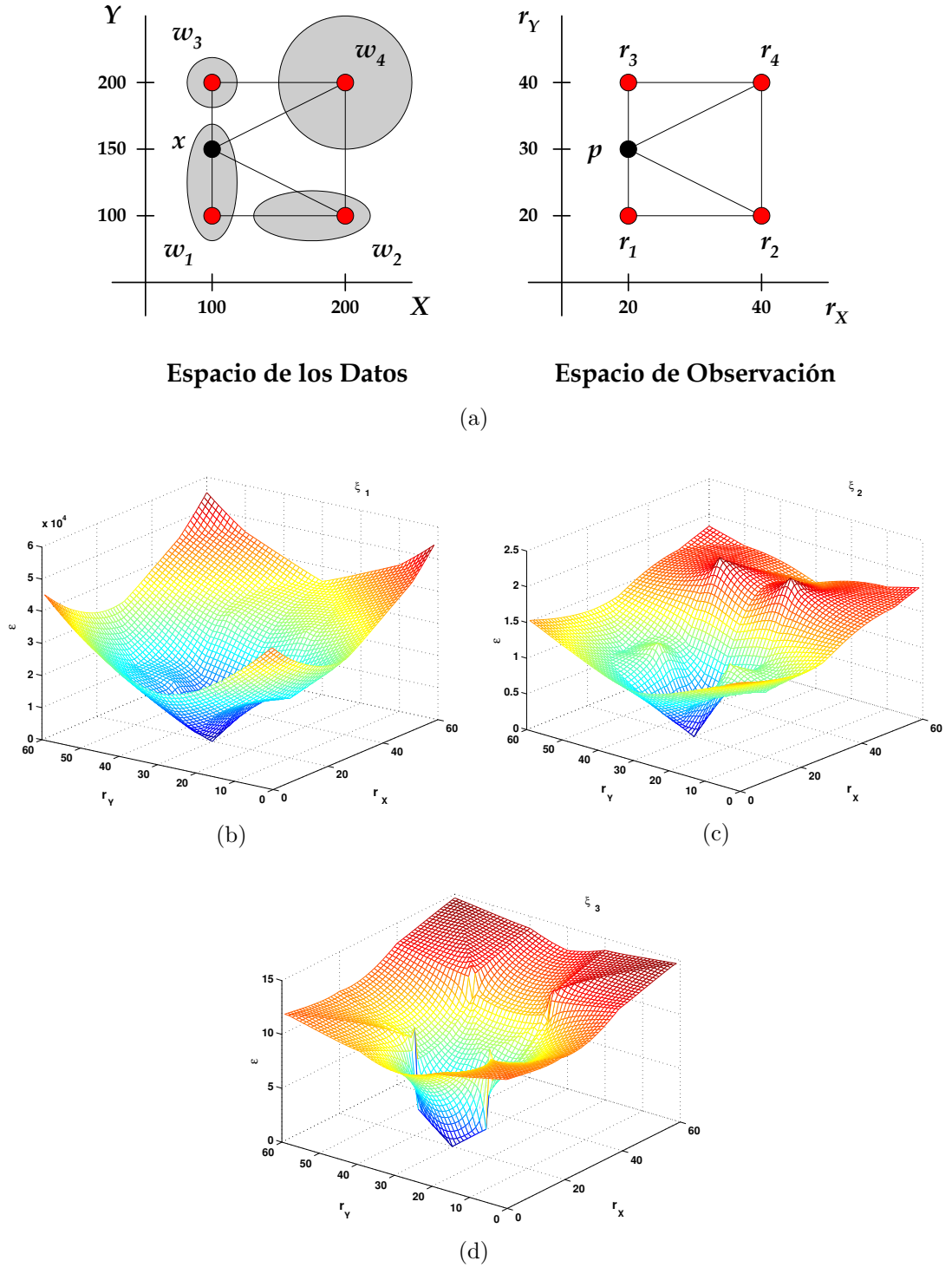


Figura 6.7: Ejemplo del MST. Un MAO  $(2 \times 2)$ -dimensional ha sido adaptado a la distribución bidimensional formada por cuatro *clusters* de igual tamaño y diferente forma. El dato  $x = (100, 150)$  debe ser proyectado al punto  $p = (20, 30)$  (a). Las imágenes (b), (c) y (d) muestran  $\varepsilon_x(p)$  para  $\varphi = 1$  y  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ , respectivamente, donde el mínimo de estas funciones indica la correcta proyección final de  $x$ .

de una vecindad  $D$  del dato a proyectar en el espacio de los datos.

$$\Upsilon(x) = \{\eta_i \mid \|x - w_i\| \leq D \quad i = 1, \dots, M\} \quad (6.9)$$

En los experimentos  $D$  puede ser el diámetro del conjunto de datos, para asegurar (prácticamente) que  $\Upsilon(x)$  contenga a todas las neuronas del MAO; y también la mínima distancia tal que la cardinalidad de  $\Upsilon(x)$  sea mayor o igual a tres, para que  $\Psi(x)$  siempre contenga más de un par de neuronas.

- Considerar las neuronas que se encuentren dentro de una vecindad  $d$  de la neurona ganadora en el mapa.

$$\Upsilon(x) = \{\eta_i \mid \|r_c - r_i\| \leq d \quad i = 1, \dots, M\} \quad (6.10)$$

donde  $r_c$  es la posición de la neurona ganadora asociada al dato de entrada  $x$ . En los experimentos generalmente  $d = A\sqrt{2}$ , donde  $A$  es la separación horizontal o vertical entre dos neuronas en la rejilla regular, de forma que sólo las ocho (como máximo) neuronas adyacentes a la ganadora sean consideradas.

En ambos casos la vecindad está definida mediante la distancia Euclídea, aunque es posible utilizar otras medidas como la vecindad generada por el MAO de árbol generador mínimo, véase la Sección 3.3.1.3.

El siguiente paso es definir las parejas de neuronas pertenecientes a  $\Upsilon(x)$  que van a formar parte del conjunto  $\Psi(x)$ . Los experimentos utilizan las dos siguientes estrategias:

- Considerar todos los  $u(u-1)/2$  pares posibles con las  $u$  neuronas de  $\Upsilon$ .

$$\Psi(x) = \{(\eta_i, \eta_j) \mid i < j \quad \forall \eta_i, \eta_j \in \Upsilon(x)\} \quad (6.11)$$

- Considerar los  $u-1$  pares contruidos con la neurona ganadora  $\eta_c$  y otra neurona de  $\Upsilon$ .

$$\Psi(x) = \{(\eta_c, \eta_j) \mid c \neq j \quad \forall \eta_j \in \Upsilon(x)\} \quad (6.12)$$

donde  $\eta_c$  es la neurona ganadora fija asociada al dato  $x$ .



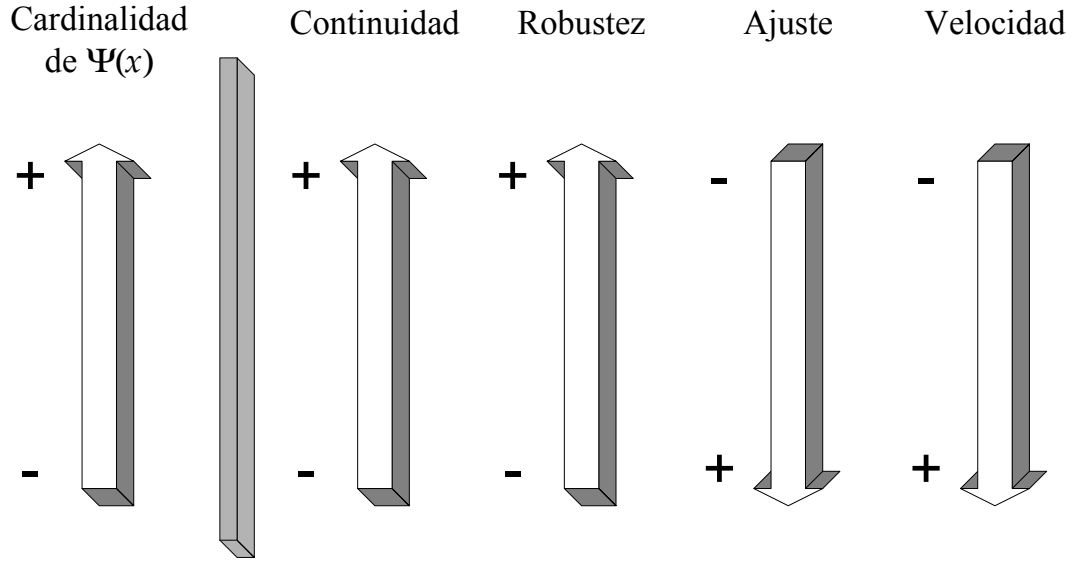


Figura 6.8: Relación entre las propiedades de la proyección del método y la cardinalidad del conjunto de pares de neuronas  $\Psi(x)$ .

De esta manera, queda definida la cardinalidad de  $\Psi(x)$  que resulta ser clave de cara a satisfacer las propiedades de la función  $f$ , véase la Sección 6.1.1. La Figura 6.8 ilustra la relación entre la cardinalidad de  $\Psi(x)$  y las propiedades de la proyección, suponiendo que los vectores de referencia de los pares de neuronas se hallan próximos al dato a proyectar y que las neuronas del MAO están adecuadamente ordenadas. Los métodos de interpolación descritos en la literatura también se ajustan a este esquema. Al considerar pocas neuronas son rápidos y satisfacen la propiedad de ajuste, aunque no proporcionan una proyección continua ni robusta. El MST otorga más flexibilidad a la hora de escoger un conjunto de neuronas de referencia permitiendo al analista controlar el grado con el que las propiedades son satisfechas.

El siguiente apartado describe varios experimentos llevados a cabo con el algoritmo, mientras la Sección 6.1.4 presenta una discusión sobre los resultados obtenidos.

### 6.1.3. Resultados experimentales

Este apartado describe varios experimentos con datos de simulación. El objetivo consiste en ilustrar las características del MST, analizar sus ventajas y

limitaciones, y compararlo con otros métodos. Los tiempos de ejecución presentados son relativos a un ordenador personal con procesador Intel® Pentium® 4 a 1.7GHz con 256MB de memoria RAM. La Sección 6.3.3.17 muestra varias proyecciones del método a partir de un MAO adaptado a datos reales de habla.

#### 6.1.3.1. MAO $(2 \times 2)$ -dimensionales + distribuciones bidimensionales

Dado que la mayoría de MAO utilizados para AED son bidimensionales, el primer objetivo planteado a la hora de construir la función de proyección  $f$  fue que fuese capaz de proyectar adecuadamente distribuciones bidimensionales. En otras palabras, el método debe ser capaz de generar una función  $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$  que permita visualizar adecuadamente una distribución de datos bidimensional.

La Figura 6.9 muestra un primer ejemplo básico del método donde dos MAO  $(2 \times 2)$ -dimensionales han sido adaptados a una circunferencia. Cuando el orden de los vectores de referencia en el espacio de los datos coincide con el de las neuronas en la rejilla regular del mapa la proyección del MST revela la esperada forma circular (a), ya que la topología es preservada correctamente. Sin embargo, el MAO puede “doblar” de manera que dos vectores de referencia intercambien sus posiciones relativas (en la imagen, los representados mediante azul y rojo), en cuyo caso el MAO genera la esperada proyección en forma de “ $\infty$ ” (b). La Figura 6.10 muestra un MAO  $(2 \times 2)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, y la correspondiente proyección del MST que describe adecuadamente la distribución de datos original. En ambos ejemplos el error entre triángulos es  $\xi_1$ ,  $\varphi = 1$  y  $\Psi(x)$  contiene todos los posibles pares de neuronas formados con las cuatro neuronas del MAO. De esta manera, el método es capaz de resolver las deficiencias del método de interpolación basado en transformaciones afines generando proyecciones continuas, donde tampoco aparece el fenómeno de “inversión” de las clases, véase la Figura 5.28.

Una vez visto que el método es capaz de representar distribuciones bidimensionales para el caso más sencillo (con los MAO bidimensionales más pequeños), el siguiente paso es evaluar el método para mapas ligeramente más grandes. A continuación varios experimentos son descritos con MAO  $(5 \times 5)$ -dimensionales, los cuales han sido entrenados con distribuciones de diferente dimensión intrínseca.

### 6.1. MÉTODO DE SEMEJANZA DE TRIÁNGULOS

---

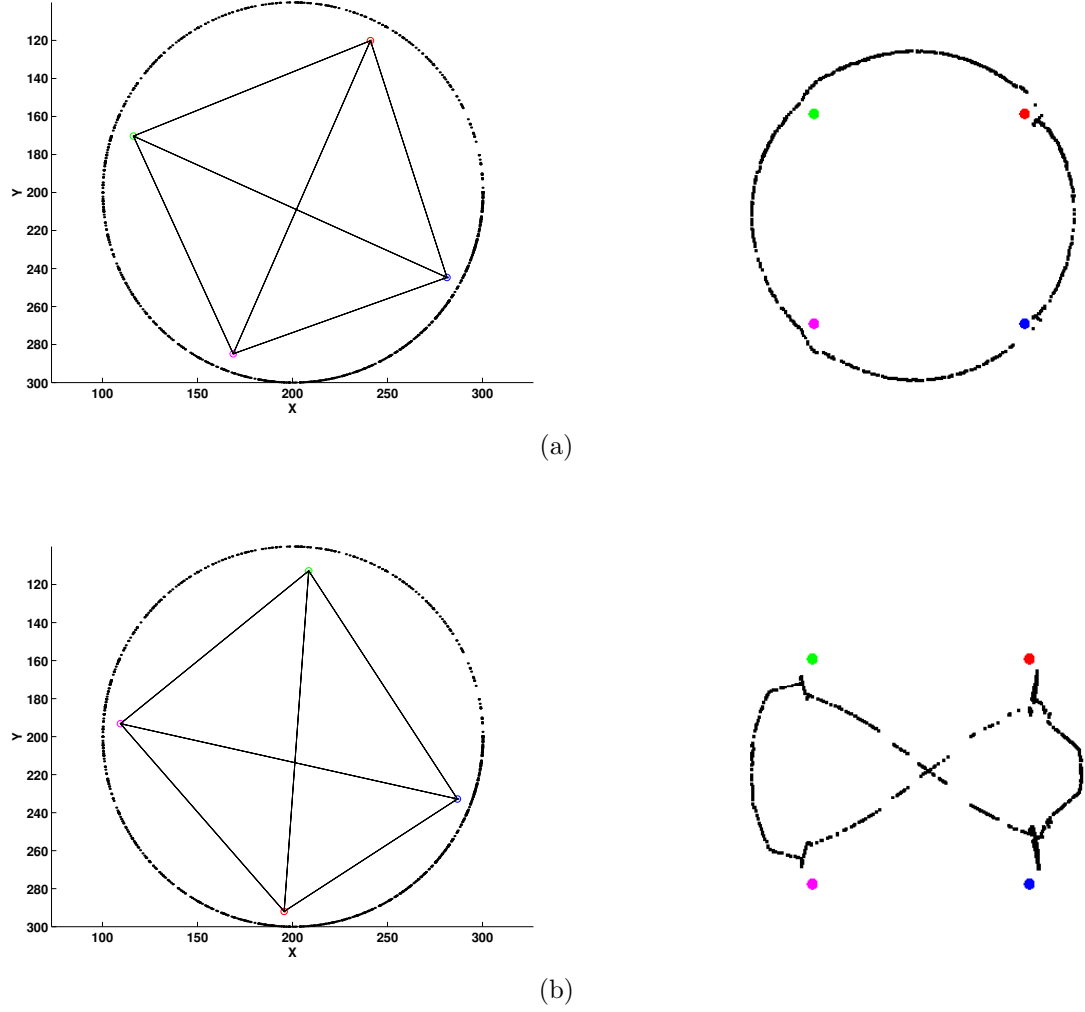


Figura 6.9: Proyecciones del MST para dos MAO ( $2 \times 2$ )-dimensionales adaptados a una circunferencia. En (a) el orden de los vectores de referencia coincide con el de las neuronas en la rejilla del mapa. En (b) el MAO se ha “doblado”, donde las posiciones relativas de los vectores de referencia azul y rojo aparecen intercambiadas.

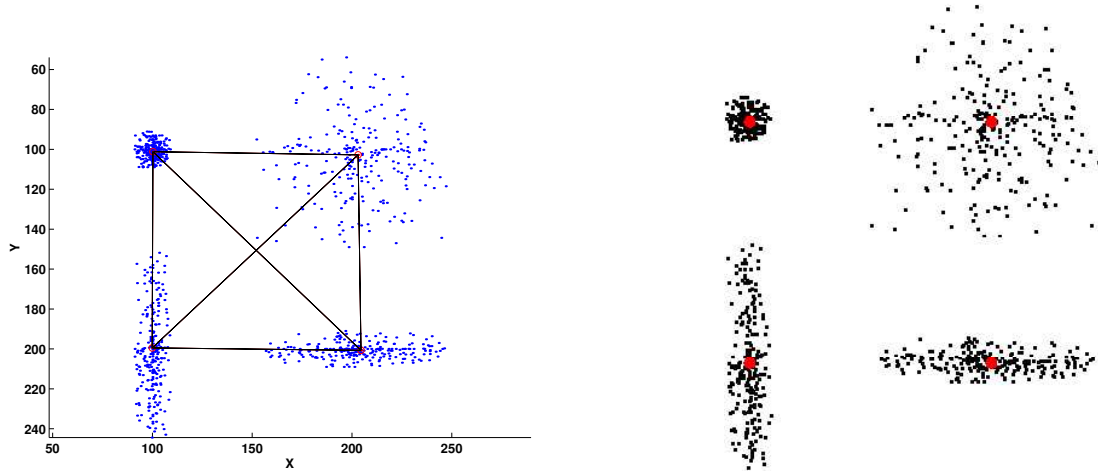


Figura 6.10: Proyección del MST para un MAO  $(2 \times 2)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

### 6.1.3.2. MAO $(5 \times 5)$ -dimensional + círculo y anillo

En este experimento el objetivo consiste en analizar la capacidad del método para representar distribuciones bidimensionales con MAO pequeños y evaluar los tres tipos de errores entre triángulos  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ .

La Figura 6.11 muestra un MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por un círculo y un anillo (a). En general, resulta muy difícil adaptar el MAO de modo que preserve perfectamente la topología ya que la atracción de los vectores de referencia que representan al círculo suele desplazar algún vector de referencia, que “debería” ajustarse al anillo, hacia el centro de la distribución. De esta manera, es posible observar que el MAO se dobla (en forma de “C”) para ajustarse a la distribución, apareciendo un “hueco” en su parte superior donde no existe una conexión entre los vectores de referencia ajustados al anillo. Por tanto, cabe esperar que la proyección del MST ilustre un *cluster* central rodeado por un anillo abierto. Las imágenes (b), (c) y (d) muestran las proyecciones del MST para  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ , respectivamente, donde  $\Psi(x)$  está compuesto por todos los posibles pares de neuronas. Las imágenes (e), (f) y (g) muestran las proyecciones para  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ , respectivamente, utilizando un conjunto reducido de pares de neuronas (en concreto,  $\Psi(x)$  contiene todos los pares de neuronas cuyos vectores de referencia se encuentran a una cierta distancia  $D$  del dato a proyectar, siendo  $D$  la mínima distancia tal que la cardinalidad de  $\Upsilon(x)$  es siempre mayor o igual a tres). En todos los casos  $\varphi$  es escogida según (6.6).

Las imágenes generadas a partir del error  $\xi_2$  no ilustran las características de

## 6.1. MÉTODO DE SEMEJANZA DE TRIÁNGULOS

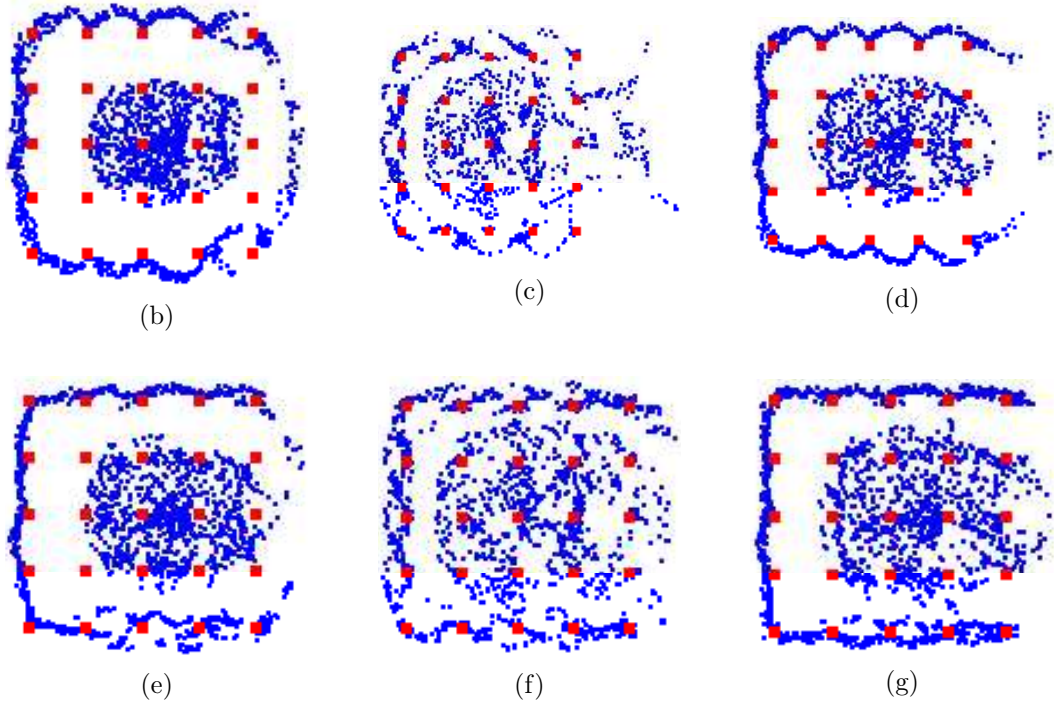
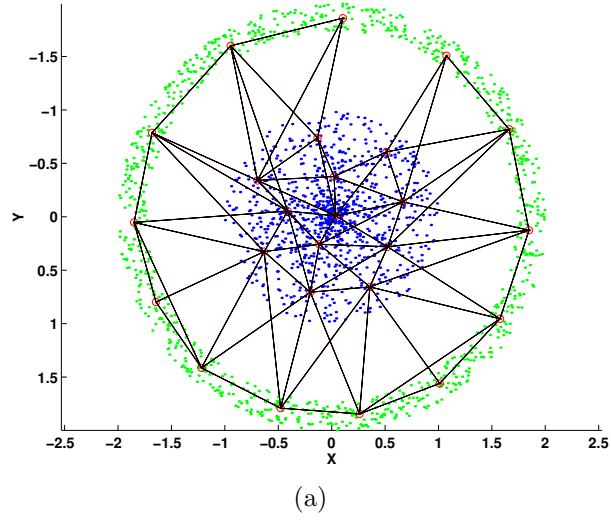


Figura 6.11: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por un círculo y un anillo (a). Las imágenes (b), (c) y (d) muestran las proyecciones del MST para  $\xi_1$ ,  $\xi_2$  y  $\xi_3$ , respectivamente, donde intervienen todos los posibles pares de neuronas para construir la proyección. Las imágenes (e), (f) y (g) muestran las proyecciones utilizando un conjunto reducido de pares de neuronas. En general,  $\xi_1$  aporta mejores resultados.

Figura 6.11	(b)	(c)	(d)	(e)	(f)	(g)
Tiempo de ejecución (segundos)	252	514	215	75	107	69

Tabla 6.1: Tiempos de ejecución para generar las imágenes de la Figura 6.11.

Figura 6.13	(a)	(b)	(c)	(d)
Tiempo de ejecución (segundos)	6.53	0.83	1.81	0.99

Tabla 6.2: Tiempos de ejecución para generar las imágenes de la Figura 6.13.

la distribución tan claramente como las creadas a partir de  $\xi_1$  y  $\xi_3$ , a pesar de que con  $\xi_2$  se satisface la propiedad de ajuste e introduce una modificación a  $\xi_1$  de modo que resulta invariante ante el tamaño de los triángulos, como sucede con  $\xi_3$ . Por otro lado, las primeras proyecciones (b) (c) y (d) en las que interviene un mayor número de pares de neuronas producen generalmente descripciones más precisas y fiables de las distribuciones de datos al ser continuas. Sin embargo, la velocidad del método disminuye a medida que más pares de neuronas son considerados. La Tabla 6.1 ilustra los tiempos de ejecución aproximados para generar las seis proyecciones de la Figura 6.11. Por otro lado, las imágenes creadas a partir de  $\xi_2$  generan proyecciones más dispersas, lo cual conlleva una prolongación del tiempo de ejecución dado el algoritmo de optimización utilizado. En general, los tiempos de ejecución pueden ser notablemente reducidos empleando un algoritmo de optimización más eficiente (véase la Sección 6.1.2).

Con respecto a la diferencia entre  $\xi_1$  y  $\xi_3$ , con el primer error el anillo exterior queda ligeramente mejor representado, ya que aparece un mayor grado de continuidad en el arco del anillo del lateral derecho de las proyecciones, indicando la forma cíclica o cerrada de la distribución. En este sentido,  $\xi_1$  es más robusto a la adaptación del MAO a la distribución. De hecho, la propiedad de ajuste se cumple perfectamente con  $\xi_2$  y  $\xi_3$ , mientras que para  $\xi_1$  sólo se cumple para las neuronas asociadas al anillo exterior, dando más libertad a la proyección para representar el arco exterior derecho.

Experimentalmente se observa que la propiedad de ajuste es satisfecha con todas las distribuciones de simulación y MAO ( $5 \times 5$ )-dimensionales cuando el error empleado es  $\xi_2$ . En cambio, con  $\xi_3$  presenta algo más de dificultad para satisfacer la propiedad, mientras que con  $\xi_1$  rara vez la satisface. Con MAO de mayor tamaño sucede algo similar. Con  $\xi_2$  un elevado número de vectores de referencia son proyectados sobre la posición de sus respectivas neuronas. Este número decrece generalmente al utilizar  $\xi_3$ , y más aún con  $\xi_1$ . En este sentido, desde un punto de vista subjetivo las proyecciones de mayor calidad son generadas a partir de  $\xi_1$ , precisamente por su capacidad de desvincularse de la estructura regular rígida de las neuronas en el mapa.

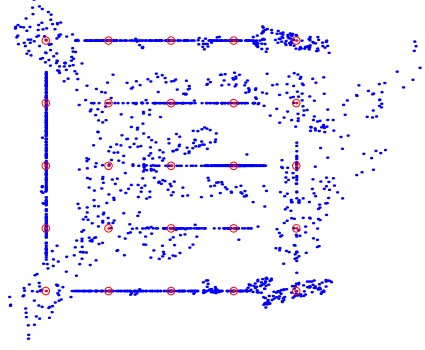


Figura 6.12: Proyección del método de interpolación basado en transformaciones afines para el MAO ( $5 \times 5$ )-dimensional y distribución ilustrados en la Figura 6.11 (a).

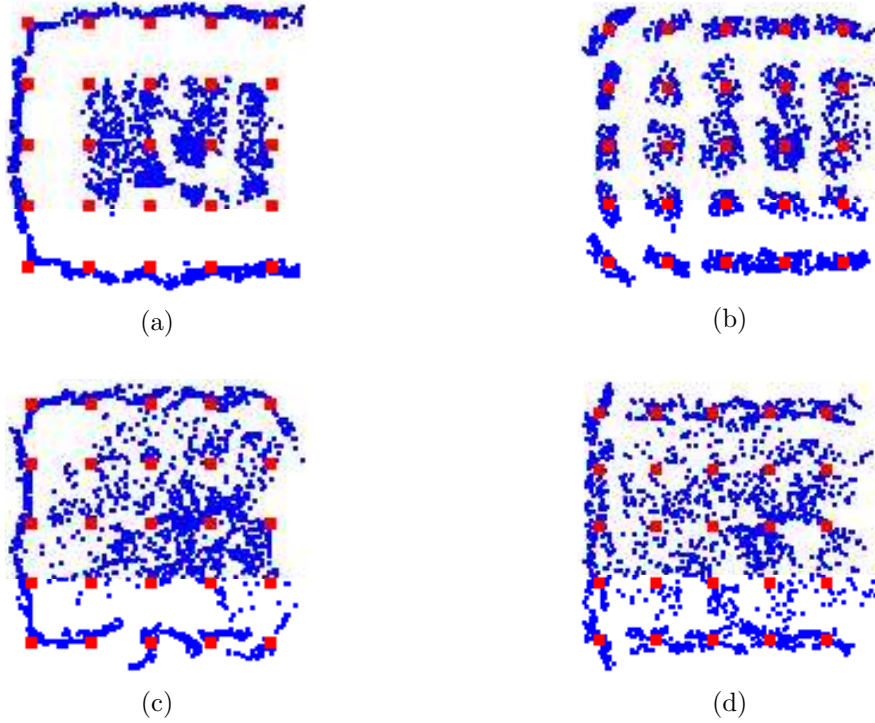


Figura 6.13: Proyecciones del MST para  $\xi_1$  con conjuntos reducidos de pares de neuronas. El MAO y distribución aparecen ilustrados en la Figura 6.11 (a).

En general,  $\xi_1$  ha aportado los mejores resultados en los experimentos, donde las imágenes que mejor describen las distribuciones de datos son generadas utilizando todos los pares de neuronas del MAO. En estos casos existe una clara mejoría con respecto al método de interpolación basado en transformaciones afines, cuya proyección es ilustrada en la Figura 6.12 para el MAO y distribución ilustrados en la Figura 6.11 (a). La ventaja principal del MST radica en la posibilidad de emplear más pares de neuronas. La Figura 6.13 muestra proyecciones del MST basadas en  $\xi_1$  y en conjuntos muy reducidos de pares de neuronas, para el mismo MAO y distribución. En (a) se han utilizado todos los pares de neuronas formados por la neurona ganadora y todas sus vecinas adyacentes. En (b) se consideran los pares formados por la neurona ganadora y una de sus vecinas, donde la proyección es similar a la de los métodos de interpolación aleatoria. En (c)  $\Psi(x)$  contiene los tres pares de neuronas formados con las tres neuronas cuyos vectores de referencia se hallan lo más cerca del dato a proyectar, y finalmente (d) considera los dos pares formados por la neurona ganadora y las dos neuronas adyacentes cuyos vectores de referencia se hallan más próximos al dato, con la restricción de que no aparezcan en línea recta en el mapa. La calidad de las proyecciones decrece a medida que el conjunto de pares de neuronas utilizados es reducido. Aunque en (a) la propiedad de ajuste no se satisface, el conjunto de neuronas ya es demasiado pequeño como para representar adecuadamente el anillo exterior (en el resto de imágenes si se satisface la propiedad). Por el contrario, la velocidad del algoritmo es superior al disminuir la cardinalidad de  $\Psi(x)$ . La Tabla 6.2 ilustra los tiempos de ejecución del algoritmo para generar las imágenes de la Figura 6.13.

#### 6.1.3.3. MAO ( $5 \times 5$ )-dimensional + senoide

Este apartado presenta otro experimento para analizar las características generales del método, el cual analiza la proyección del MST basado en el error  $\xi_1$  para un MAO ( $5 \times 5$ )-dimensional adaptado a un periodo de una senoide.

La Figura 6.14 muestra un MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por un periodo de una senoide (a). La imagen (b) muestra la proyección que genera el método de interpolación basado en transformaciones afines, donde se satisface la propiedad de ajuste y es posible apreciar de forma global la estructura de la distribución, a pesar de las discontinuidades y deficiencias locales en torno a las neuronas. Las imágenes siguientes muestran proyecciones del método donde



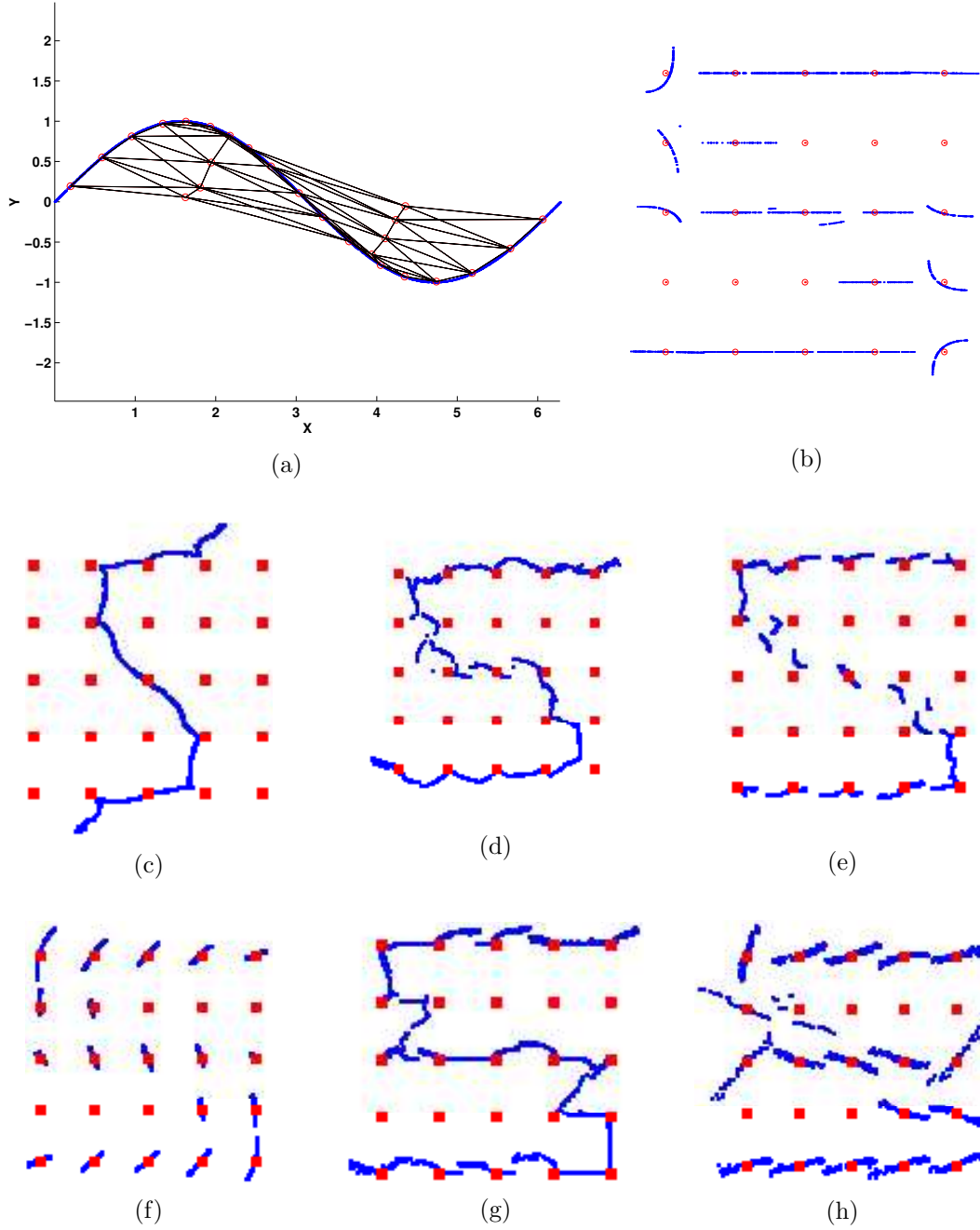


Figura 6.14: MAO ( $5 \times 5$ )-dimensional adaptado a un periodo de una senoide (a). Proyección del método basado en transformaciones afines (b). El resto de imágenes corresponden a proyecciones del MST basado en  $\xi_1$ , donde el número de pares de neuronas utilizado es reducido progresivamente (c-h).

Figura 6.14	(c)	(d)	(e)	(f)	(g)	(h)
Tiempo de ejecución (segundos)	400	72	4	0.44	1.37	0.76

Tabla 6.3: Tiempos de ejecución para generar las imágenes de la Figura 6.14.

el número de pares de neuronas utilizado es reducido progresivamente. En (c),  $\Psi(x)$  contiene todos los pares de neuronas creando una proyección continua. La imagen resultante es la que mejor ilustra la forma de la distribución original. En (d), intervienen todos los pares de neuronas cuyos vectores de referencia se encuentran a una distancia tal que la cardinalidad de  $\Upsilon(x)$  siempre es mayor o igual a tres. En este caso la propiedad de ajuste se satisface mejor pero aparecen discontinuidades. Este efecto es aún más acusado en (e), donde se han empleado todos los pares de neuronas formados por la neurona ganadora y todas sus vecinas adyacentes. En (f),  $\Psi(x)$  contiene los pares formados por la neurona ganadora y una de sus vecinas. En este caso la propiedad de ajuste se satisface para todas las neuronas, apareciendo una imagen similar a (b). En (g), se consideran los tres pares de neuronas formados con las tres neuronas cuyos vectores de referencia se hallan lo más cerca del dato a proyectar. En este caso se satisface la propiedad de ajuste y la imagen, que es generada en muy poco tiempo (véase la Tabla 6.3), aparece con un grado aceptable de continuidad. Esta proyección es interesante desde el punto de vista de analizar cómo el MAO se ha adaptado a la distribución. Finalmente, en (h) intervienen los dos pares formados por la neurona ganadora y las dos neuronas adyacentes cuyos vectores de referencia se hallan más próximos al dato, con la restricción de que no aparezcan en línea recta en el mapa, donde la imagen vuelve a ser similar a (b).

La Tabla 6.3 muestra los tiempos de ejecución para generar las imágenes de la Figura 6.14. En este ejemplo (y en la mayoría) la mejor opción de cara a obtener una representación fiel de la distribución consiste en utilizar todos los pares de neuronas. Sin embargo, la proyección de la imagen (g) es generada mucho más rápidamente y resulta ser suficientemente adecuada como para representar la distribución de datos, respetándose la propiedad de ajuste. Como conclusión, dada la rapidez de ejecución del algoritmo para conjuntos reducidos de pares de neuronas el analista puede examinar primero estas proyecciones en las que se satisface la propiedad de ajuste, antes de comenzar a analizar proyecciones más lentas basadas en grandes conjuntos de pares de neuronas.

#### 6.1.3.4. MAO ( $5 \times 5$ )-dimensionales + circunferencia

El siguiente experimento analiza la robustez del método. Para ello varios MAO ( $5 \times 5$ )-dimensionales han sido entrenados con datos procedentes de una circunferencia y con distintos parámetros de entrenamiento, de manera que presentan diferentes errores de cuantización. La Figura 6.15 muestra tres MAO y las respectivas proyecciones del MST utilizando  $\xi_1$ , para casos en los que el error de cuantización es bajo, medio y elevado, en las imágenes (a), (b) y (c), respectivamente. En el primer caso, el MAO se adapta relativamente bien a la distribución. La proyección sólo considera los tres pares de neuronas formados por las tres neuronas cuyos vectores de referencia se hallan lo más cerca de un dato de entrada, pero es capaz de describir adecuadamente la distribución. En este caso se satisface la propiedad de ajuste, por lo que es posible ver exactamente qué neuronas han sido ajustadas a la distribución (las neuronas interpolantes forman un “L” invertida en el centro del mapa) y la manera en que lo han hecho, donde es posible apreciar la continuidad de la distribución y su forma cíclica. Aunque no ha sido representada en la figura, utilizando todos los pares de neuronas es posible generar una proyección más redondeada donde la propiedad de ajuste no es satisfecha. En los siguientes casos (b) y (c) se ha optado por utilizar todos los pares de neuronas para generar la proyección, ya que un elevado porcentaje de neuronas no han sido adaptadas adecuadamente a la distribución, con lo cual la propiedad de ajuste puede ser ignorada. El segundo MAO sólo adapta 12 de sus neuronas exteriores (menos de la mitad) a la circunferencia, donde la propiedad de ajuste no llega a ser satisfecha completamente, aunque los vectores de referencia son proyectados a lugares muy próximos a las neuronas. En este caso la proyección también muestra la estructura cíclica de la distribución, qué neuronas han sido adaptadas a la circunferencia y es posible intuir una estructura circular de los datos, al observar arcos en la proyección entre cada neurona adaptada a la distribución. Finalmente en (c) la proyección ilustra claramente una estructura circular a pesar de que el MAO no se ha adaptado a la distribución. En este caso, es posible generar la proyección ya que los vectores de referencia del MAO están ordenados, y es posible considerar que se preserva la topología. Esto indica que el método puede funcionar con una serie de puntos fijos en el plano de observación diferentes a las neuronas de un MAO, siempre que éstos presenten cierto orden. En otras palabras, el método puede ser utilizado con otra correspondencia discreta  $f_d$  distinta a la que generan los MAO. Por ejemplo, el método puede estar

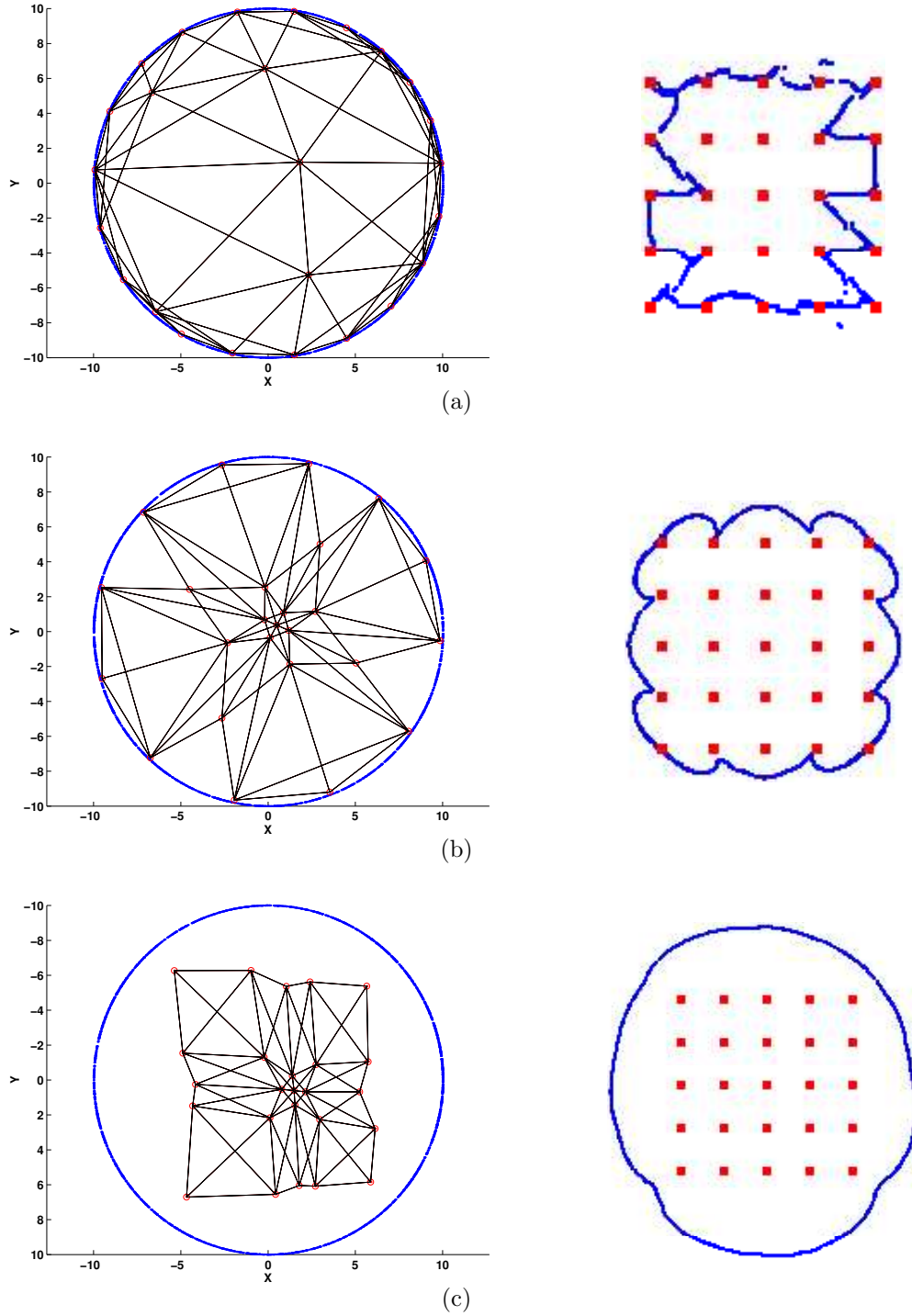


Figura 6.15: Robustez del método. Tres MAO ( $5 \times 5$ )-dimensionales adaptados a una circunferencia con error de cuantización bajo (a), medio (b), y elevado (c), a la izquierda en las imágenes. Las proyecciones del MST (derecha) ilustran la estructura cíclica de la distribución.

basado en la correspondencia discreta entre una serie de puntos en el espacio de los datos y sus imágenes tras realizar una proyección lineal de éstos al plano de observación.

#### 6.1.3.5. MAO ( $5 \times 5$ )-dimensional + dos eslabones encadenados

Las distribuciones empleadas en los experimentos anteriores son bidimensionales. Los dos siguientes apartados examinan la capacidad del método para proyectar distribuciones tridimensionales a un plano bidimensional.

La Figura 6.16 muestra como el método es capaz de ilustrar estructuras que generalmente no pueden ser visualizadas con el resto de métodos de análisis visual de MAO. El ejemplo está basado en un MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados (a). La proyección del método de interpolación basado en transformaciones afines (b) no aporta una visión adecuada sobre la distribución. Por otro lado, dado el pequeño tamaño del MAO, los métodos tradicionales como la U-matrix (c) o los histogramas de datos (en este caso la variante generada según la Tabla 5.2) (d) no aportan imágenes en las que aparezcan las dos estructuras circulares. Una posible alternativa es mostrada en (e) donde un histograma de datos generado a partir de la inserción artificial de neuronas entre las originales (según la Tabla 5.3) es capaz de discriminar las dos clases. Aumentando el tamaño del MAO también es posible detectar dos clases pero no su estructura cíclica, véase la Figura 5.6. El MST es capaz de revelar la estructura cíclica de los dos anillos al crear proyecciones continuas cuando intervienen todos los pares de neuronas. En (f),  $\Psi(x)$  contiene los pares formados por la neurona ganadora y todas sus vecinas, el error entre triángulos corresponde a  $\xi_1$  y  $\varphi$  es escogida según (6.6), donde la información aportada es similar a la imagen (e). En (g), se ha utilizado  $\xi_2$  y todos los pares de neuronas cuyos vectores de referencia se encuentran a una distancia tal que la cardinalidad de  $\Upsilon(x)$  es siempre mayor o igual a tres. Este caso es interesante ya que la proyección satisface perfectamente la propiedad de ajuste, sin embargo, todavía no es continua. La proyección en (h) utiliza todos los pares de neuronas y  $\xi_1$ . Ésta es continua por definición aunque en la imagen parece que el anillo inferior queda ligeramente abierto. Este efecto puede ser anulado mediante la incorporación de la función de ponderación descrita en (6.7). El resultado es una imagen que ilustra adecuadamente la forma de la distribución original en dos dimensiones, aportando una visión mucho más clara que la generada a partir de otros métodos

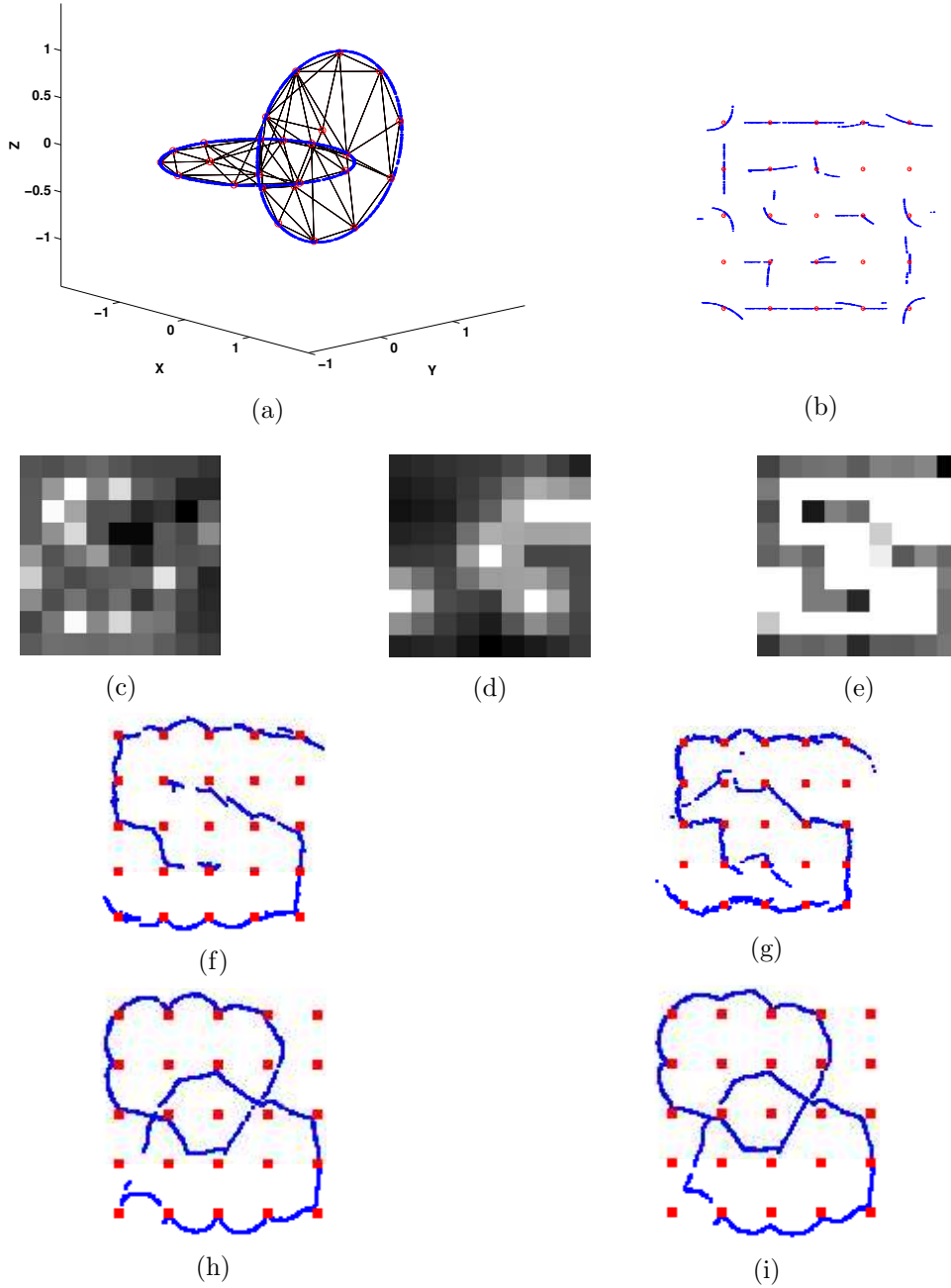


Figura 6.16: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados (a). Proyección del método basado en transformaciones afines (b). U-matrix (c). Dos variantes del histograma de datos (d) y (e). En (f) y (g) el MST utiliza conjuntos reducidos de pares de neuronas mientras que en (h) e (i) utiliza todos los pares. En (i) la función de ponderación  $\varphi$  corresponde a la descrita en (6.7) para favorecer la continuidad en la imagen.

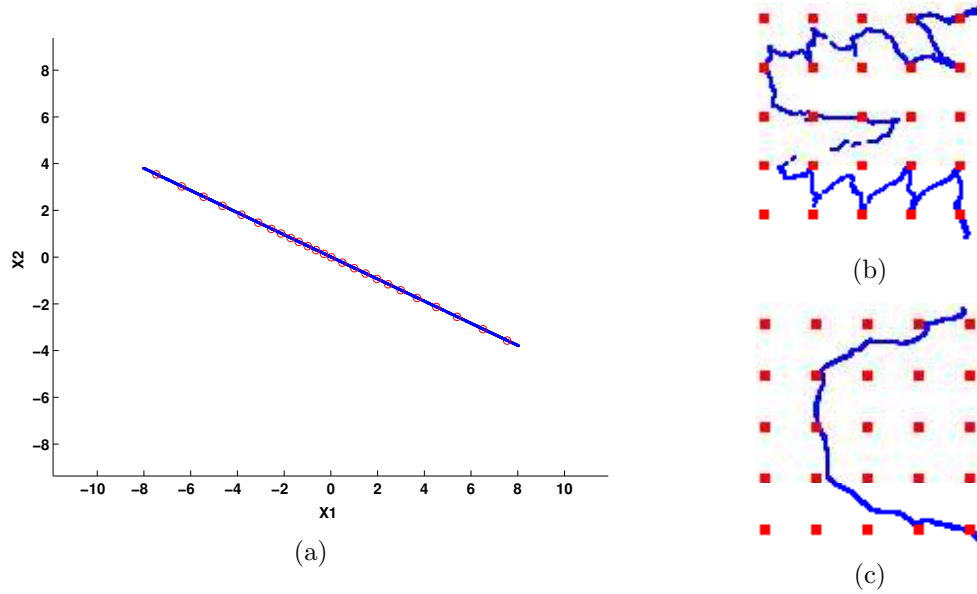


Figura 6.17: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por un segmento (a). Proyecciones del MST con un conjunto reducido de pares de neuronas (b), y con todos los pares (c).

de visualización de MAO, especialmente cuando éstos son de tamaño pequeño.

#### 6.1.3.6. Más ejemplos con MAO ( $5 \times 5$ )-dimensionales

Los siguientes tres ejemplos muestran dos proyecciones del MST basadas en  $\xi_1$  y donde  $\varphi$  es escogida según (6.7), por lo que las proyecciones son continuas por definición. La primera proyección utiliza todos los pares de neuronas cuyos vectores de referencia se encuentran a una distancia tal que la cardinalidad de  $\Upsilon(x)$  es siempre mayor o igual a tres, mientras la segunda utiliza todos los pares.

La Figura 6.17 muestra un MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por un segmento (a). Las proyecciones del método (b) y (c) permiten determinar con claridad que la dimensión intrínseca de la distribución es uno. En (b) la propiedad de ajuste no llega a satisfacerse para todas las neuronas, aunque la imagen permite visualizar como se ha adaptado el MAO a la distribución. En (c) la imagen presenta mayor continuidad y la propiedad de ajuste no se satisface para la mayoría de neuronas, originando una proyección más libre de las posiciones fijas de las neuronas en el mapa.

Lo mismo ocurre en el siguiente experimento ilustrado en la Figura 6.18, donde un MAO ( $5 \times 5$ )-dimensional ha sido adaptado a la distribución formada por dos segmentos (a). En este caso la propiedad de ajuste se satisface en (b) y es posible

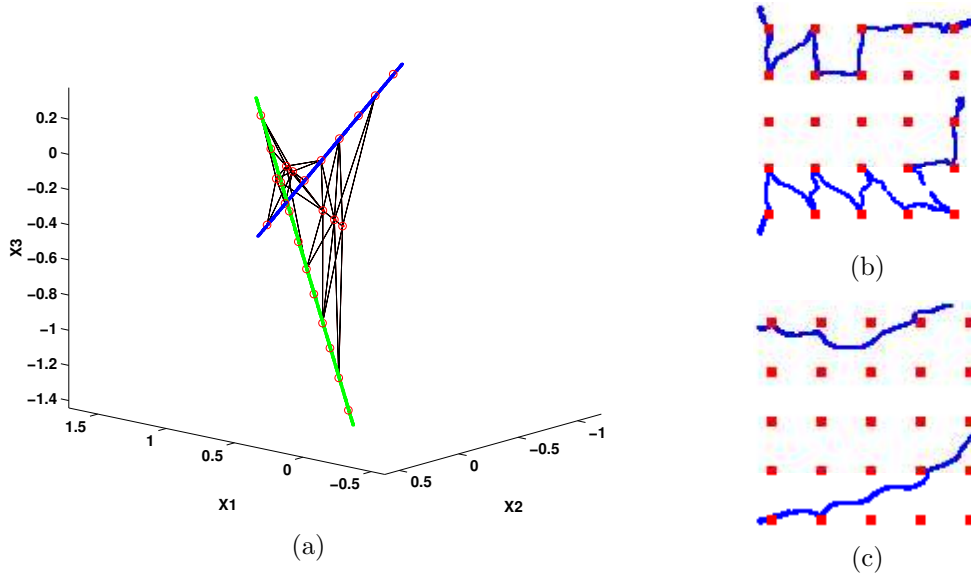


Figura 6.18: MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por dos segmentos (a). Proyecciones del MST con un conjunto reducido de pares de neuronas (b), y con todos los pares (c).

detectar dos clases de dimensión intrínseca uno tanto en (b) como en (c).

El siguiente experimento ilustrado en la Figura 6.19 es una continuación del presentado en la Figura 6.10. En este caso el MAO es  $(5 \times 5)$ -dimensional (a). A pesar de utilizar un MAO de mayor tamaño, las proyecciones (b) y (c) describen adecuadamente la distribución formada por cuatro *clusters* de igual tamaño y diferente forma. En (b) se satisface la propiedad de ajuste por lo que es posible ver exactamente cómo se han adaptado las neuronas. La proyección en (c) es más robusta y no satisface la propiedad de ajuste permitiendo visualizar con mayor precisión la forma original de los datos.

Los seis siguientes experimentos utilizan  $\xi_1$ ,  $\Psi(x)$  está compuesto por todos los pares de neuronas posibles y  $\varphi$  es escogida según (6.7).

La Figura 6.20 muestra como el método es capaz de ilustrar las cuatro clases de la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro, con un MAO pequeño, teniendo en cuenta que es imposible representar adecuadamente el ortoedro en dos dimensiones. En este caso, la imagen del MST es notablemente superior a las de otros métodos de visualización cuando el tamaño del MAO es pequeño.

La Figura 6.21 muestra otro ejemplo de una distribución difícil de analizar con métodos tradicionales de visualización de MAO cuando éstos son pequeños. La proyección muestra las seis ramas o brazos de la distribución con forma de



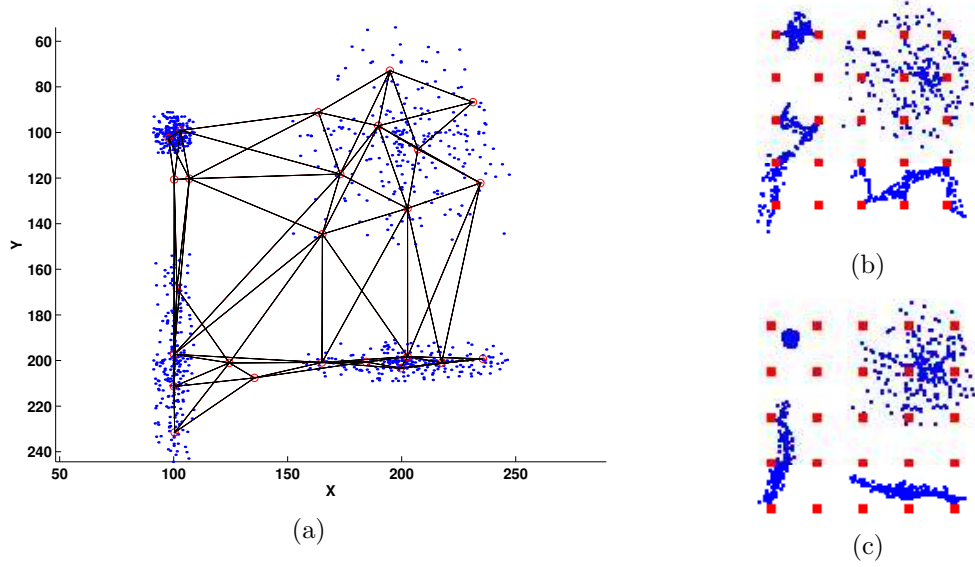


Figura 6.19: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma (a). Proyecciones del MST con un conjunto reducido de pares de neuronas (b), y con todos los pares (c).

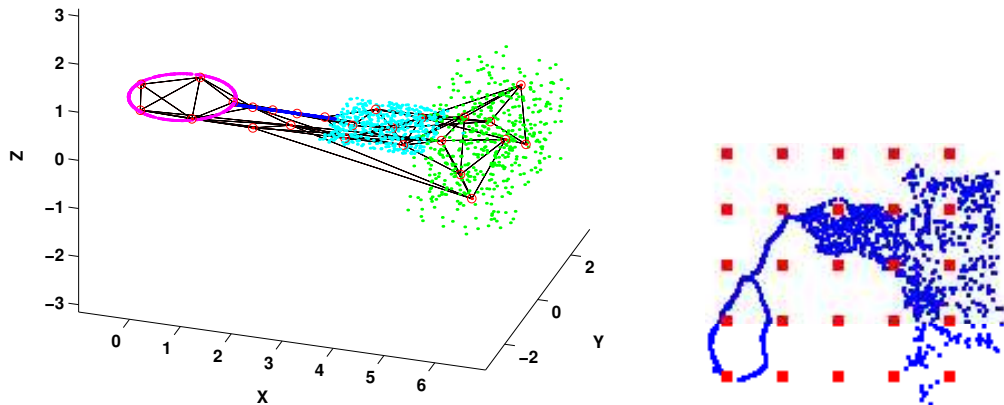


Figura 6.20: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro, y proyección del MST.

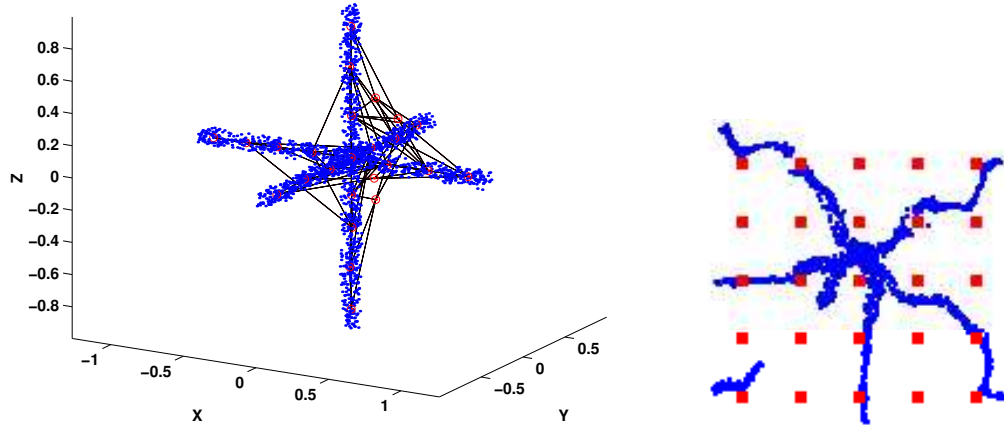


Figura 6.21: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución con forma de estrella, y proyección del MST.

estrella, donde aparece una discontinuidad evidente en la zona inferior-izquierda del MAO, a pesar de que la proyección es continua por definición. Esto es un efecto debido a la propia adaptación del MAO a la distribución tridimensional, que puede ser eliminado utilizando otro MAO que se adapte de manera más ordenada a la distribución u otro de tamaño ligeramente mayor.

La Figura 6.22 muestra la robustez del método y resulta especialmente interesante ya que, a pesar de que el MAO no puede adaptarse adecuadamente a la distribución tridimensional, la imagen resultante no es muy diferente de la que se esperaría obtener mediante otra técnica de proyección.

Los últimos tres ejemplos de este apartado presentados en las Figuras 6.23, 6.24 y 6.25 están basados en distribuciones que pueden ser tratadas por otros métodos de visualización de MAO aunque su tamaño sea relativamente pequeño (por ejemplo, las proyecciones son equivalentes a histogramas de datos). En estos casos, el MST puede aportar una visión ligeramente diferente de los datos que ayude a su interpretación y análisis.

#### 6.1.3.7. MAO ( $20 \times 20$ )-dimensionales

Los siguientes experimentos confirman que el método puede ser aplicado igualmente con MAO de mayor tamaño. Varias proyecciones del MST han sido calculadas con MAO ( $20 \times 20$ )-dimensionales adaptados a distribuciones de simulación.

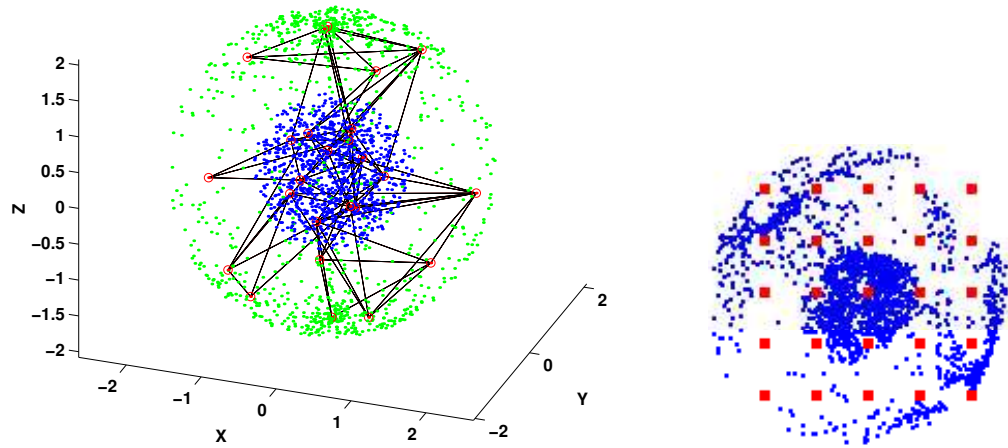


Figura 6.22: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca, y proyección del MST.

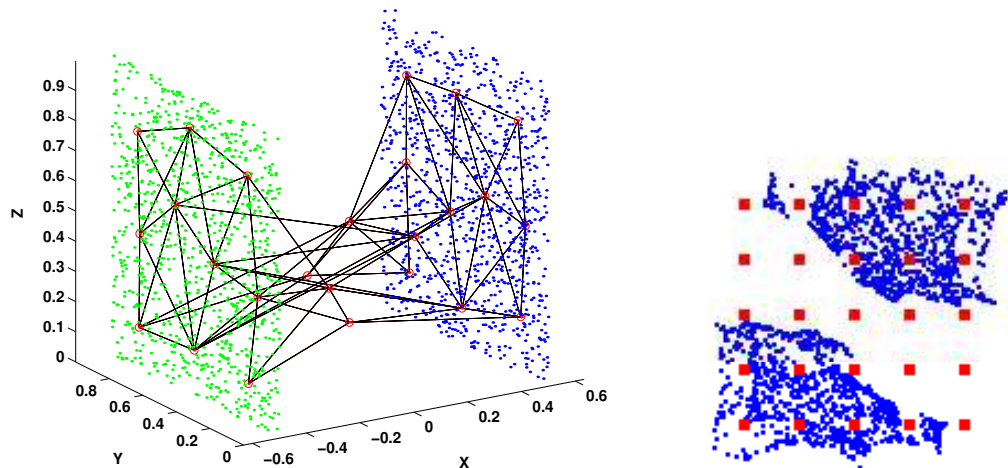


Figura 6.23: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por dos superficies planas paralelas, y proyección del MST.

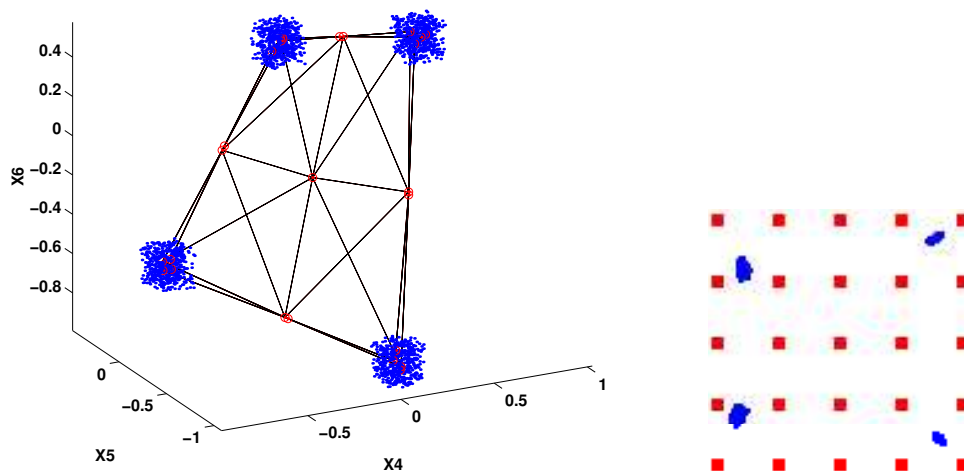


Figura 6.24: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos con forma hipercúbica, y proyección del MST.

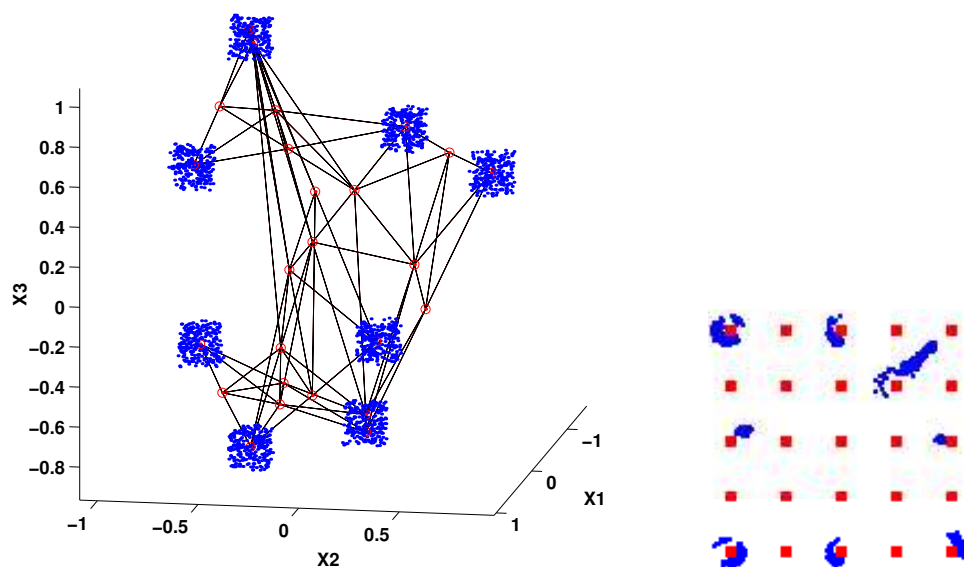


Figura 6.25: MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica, y proyección del MST.

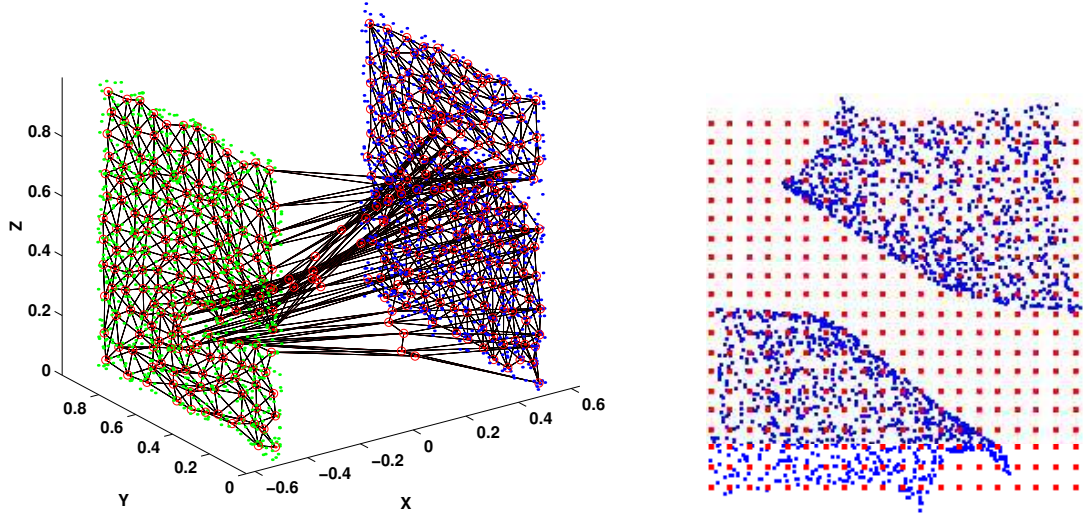


Figura 6.26: MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por 2 superficies planas paralelas, y proyección del MST.

El error entre triángulos es  $\xi_1$ ,  $\varphi$  ha sido escogida según (6.6), y  $\Psi(x)$  contiene todos los pares de neuronas posibles. Las imágenes representan adecuadamente las distribuciones, aunque el elevado número (79800) de pares de neuronas en  $\Psi(x)$  repercute negativamente en la velocidad del método (cada imagen es calculada en aproximadamente 24 horas, en el sistema descrito en la introducción de la Sección 6.1.3, aunque este tiempo puede ser reducido al menos en un orden de magnitud utilizando un algoritmo de optimización más eficiente que el empleado).

#### 6.1.4. Discusión

Un MAO únicamente aporta una correspondencia discreta entre un conjunto de vectores de referencia y otro de neuronas o posiciones fijas en un espacio observable. Por tanto, la mayoría de métodos de visualización de MAO aportan información sobre los vectores de referencia, que constituyen un conjunto generalmente reducido de representantes de los datos, en comparación con el tamaño de la distribución. La diferencia principal entre el MST y otros métodos de análisis visual de MAO reside en que éste aporta información sobre cada dato individual de la distribución mediante una visualización explícita de cada patrón. Aunque existen otros métodos que tienen en cuenta los patrones (véase la Sección 5.2), la información que aportan suele estar ligada a cada neurona o vector de referencia,

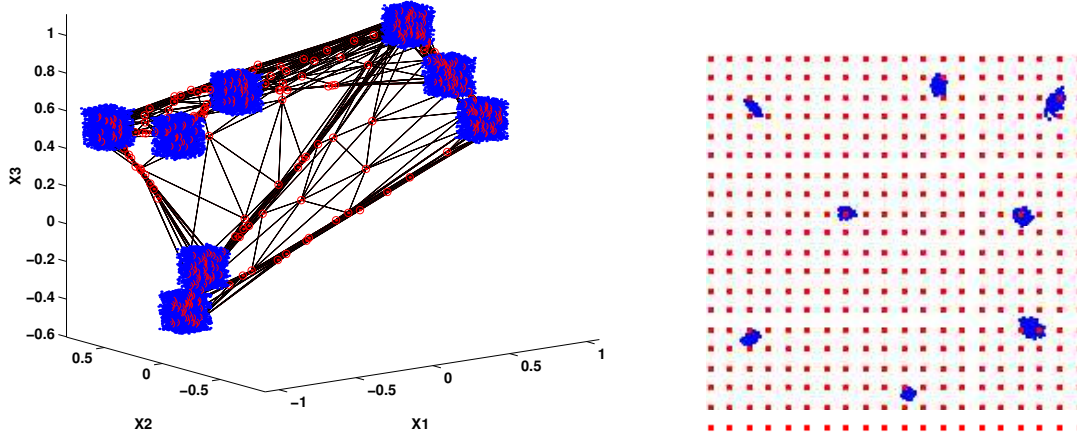


Figura 6.27: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica, y proyección del MST.

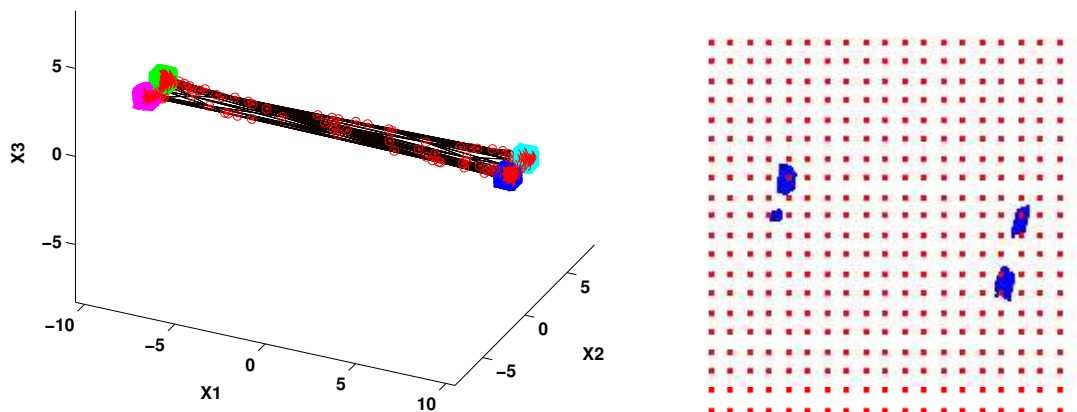


Figura 6.28: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma, y proyección del MST.



## 6.1. MÉTODO DE SEMEJANZA DE TRIÁNGULOS

---

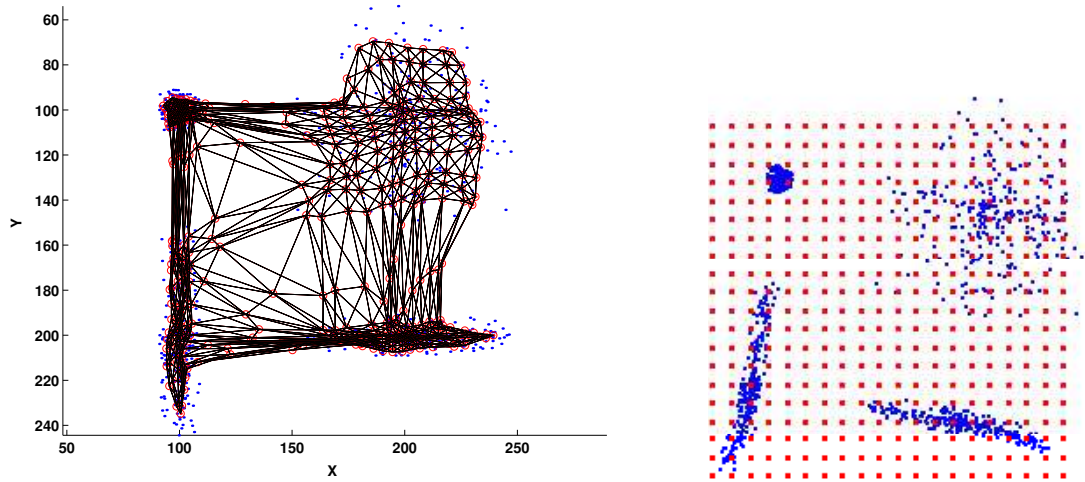


Figura 6.29: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, y proyección del MST.

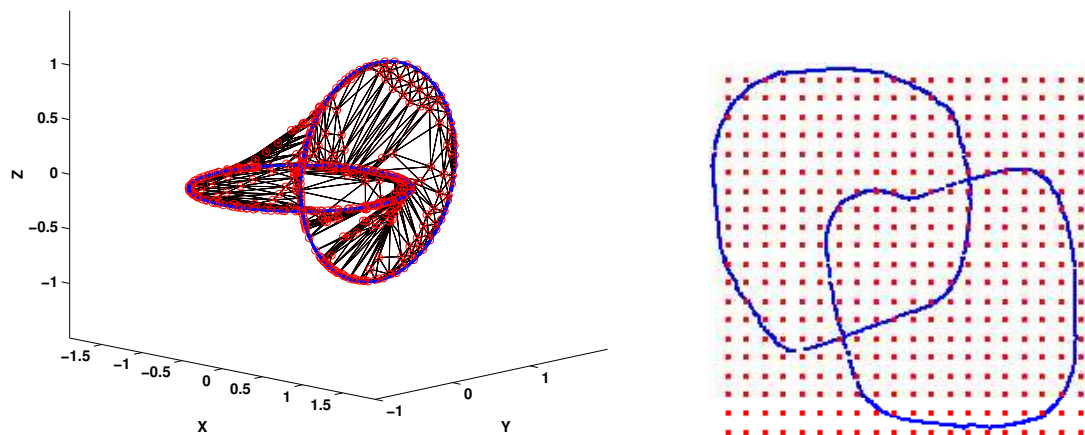


Figura 6.30: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados, y proyección del MST.

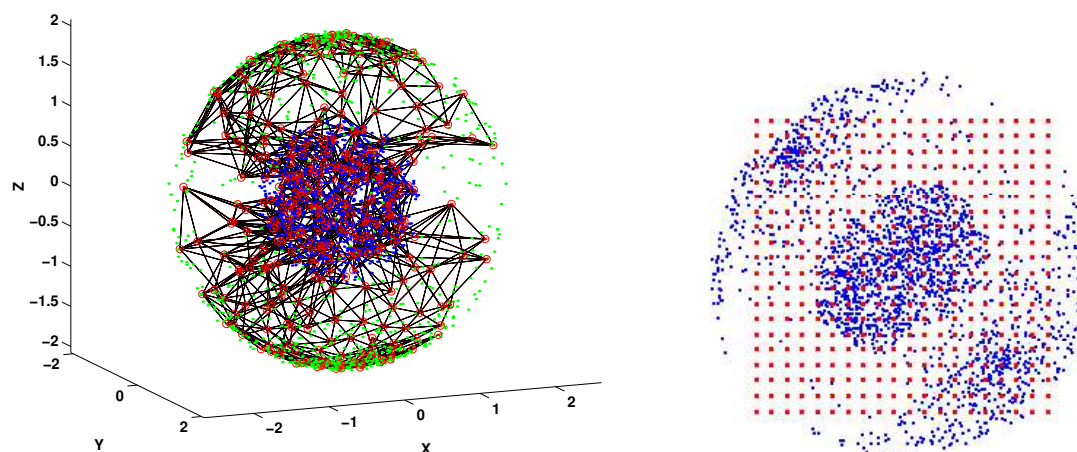


Figura 6.31: MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una esfera maciza rodeada por otra hueca, y proyección del MST.

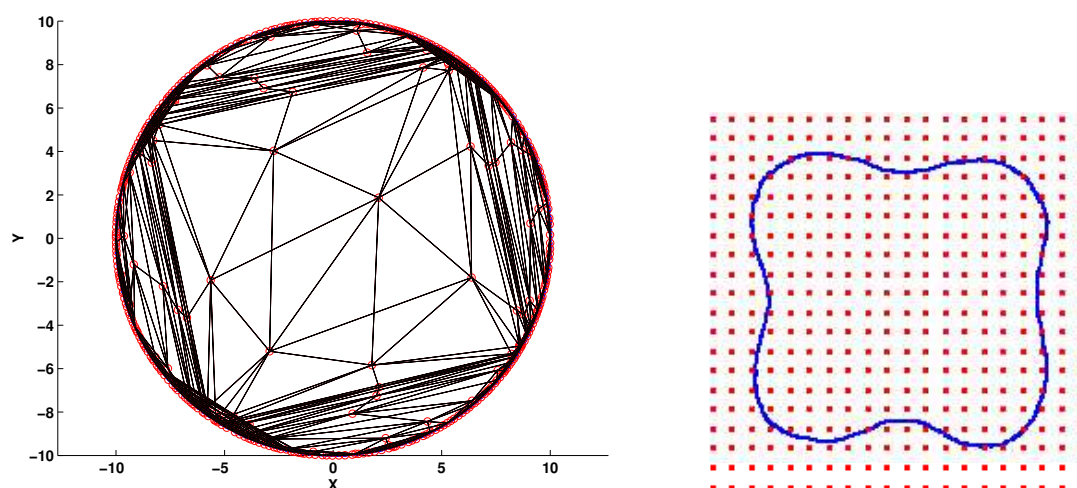


Figura 6.32: MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una circunferencia, y proyección del MST.



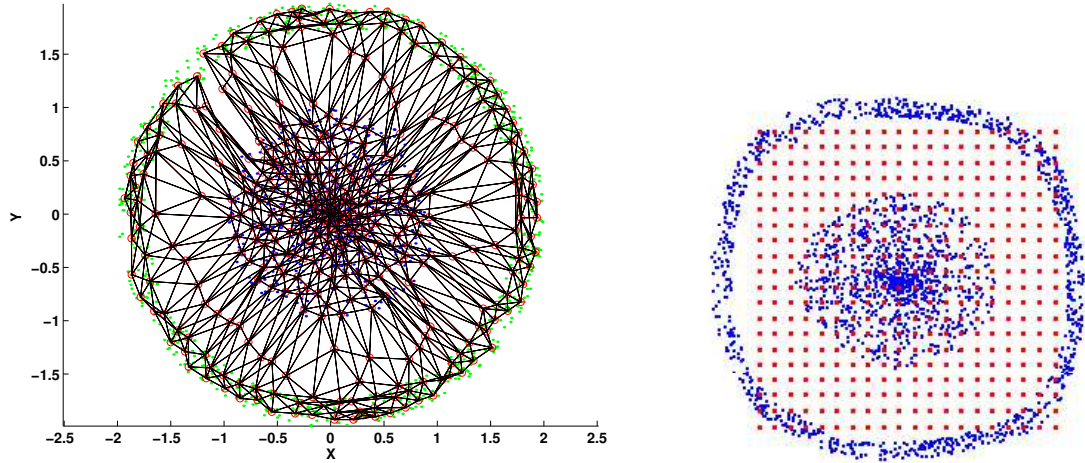


Figura 6.33: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por un círculo y un anillo grueso concéntricos, y proyección del MST.

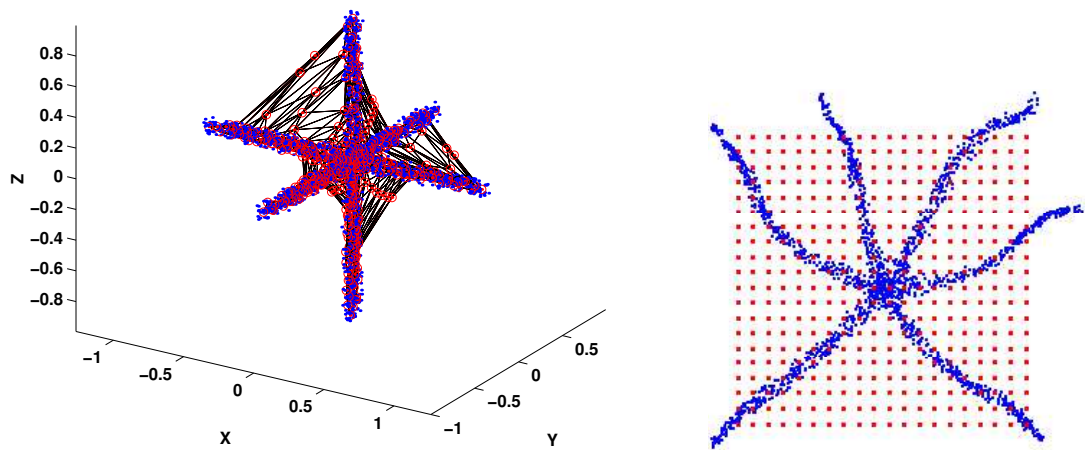


Figura 6.34: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución con forma de estrella, y proyección del MST.

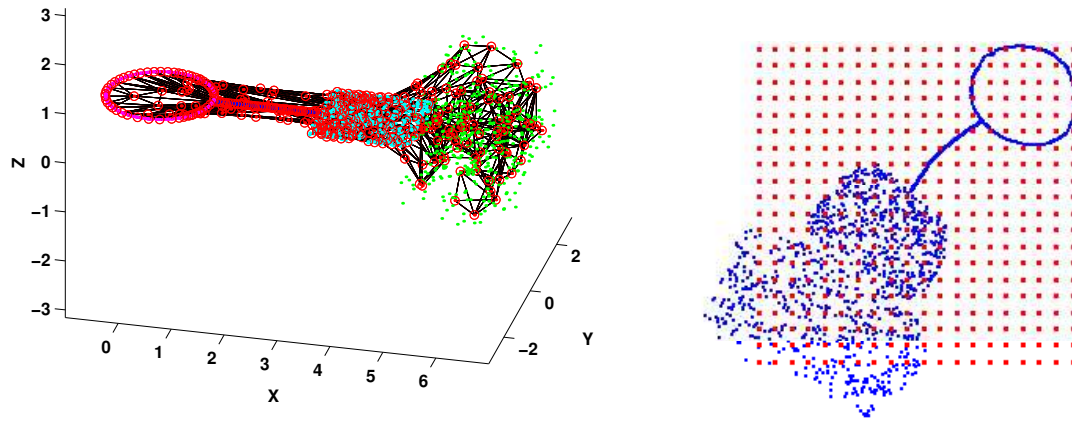


Figura 6.35: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro, y proyección del MST.

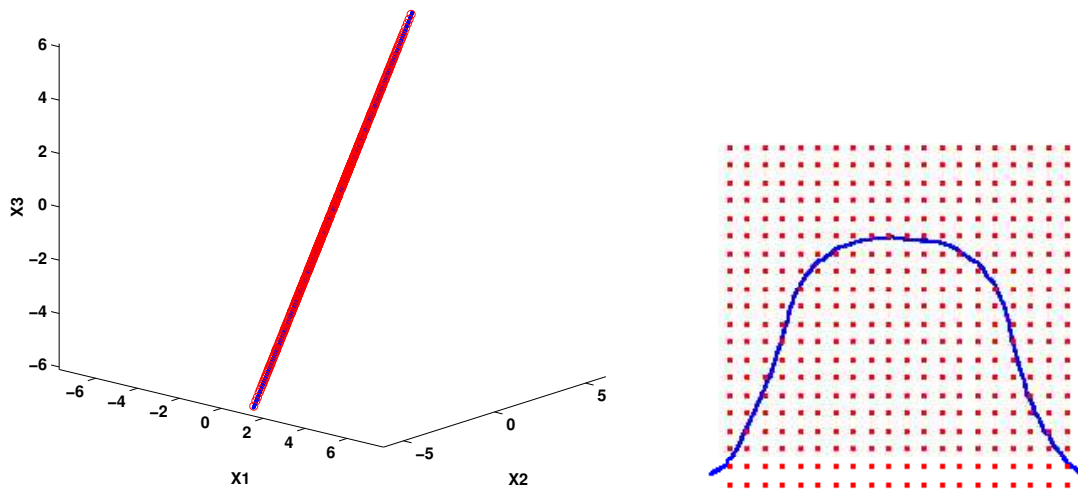


Figura 6.36: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada un segmento, y proyección del MST.

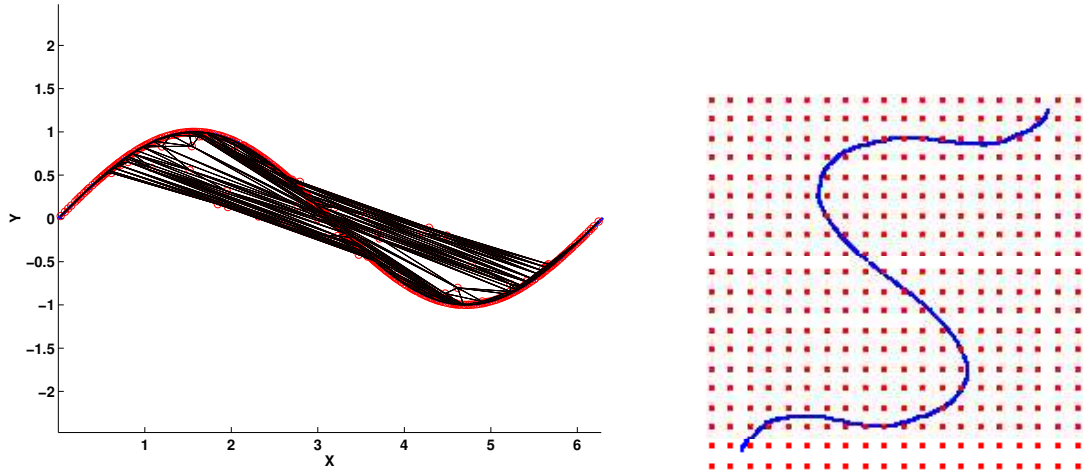


Figura 6.37: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por un periodo de una senoide, y proyección del MST.

por lo que representan características de conjuntos de datos asociados a estos representantes, y no información sobre datos individuales. En este sentido, el MST resulta atractivo desde el punto de vista del AED al poder proyectar todos los datos de la distribución de manera no-lineal.

Con respecto a otros métodos de interpolación descritos en la literatura, que también proyectan todos los datos sobre un espacio observable, su principal ventaja es la posibilidad de emplear más de tres neuronas para calcular la proyección. Esto es fundamental para generar proyecciones continuas y robustas, y aportar descripciones adecuadas de conjuntos de datos. En este sentido, es posible apreciar un aumento considerable en la calidad de las imágenes con respecto a dichos métodos de interpolación. La manera más sencilla de conseguir la continuidad consiste en emplear siempre todos los pares de neuronas formados con todas las neuronas del MAO, y utilizar una función de ponderación  $\varphi$  continua. Esta estrategia suele producir descripciones adecuadas de los conjuntos de datos. Sin embargo, al considerar todos los posibles pares la velocidad del algoritmo puede ser demasiado lenta si el tamaño del MAO no es pequeño.

El método es especialmente útil cuando el tamaño del MAO es pequeño por varias razones:

- Si éste contiene pocas neuronas el conjunto de vectores de referencia puede ser insuficiente como para representar o modelar adecuadamente la distribución de datos, por lo que es necesario utilizar un método que no tenga

una dependencia fuerte del conjunto de neuronas y sus vectores de referencia (como ocurre con el MST). En este sentido, la mayoría de métodos de visualización de MAO presentan problemas cuando éstos son pequeños. Con respecto a los métodos descritos en la Sección 5.5, que generan proyecciones no-lineales para obtener una configuración de las neuronas (diferente a la regular) en un espacio observable, y donde el conjunto de neuronas debe ser lo suficientemente grande como para describir con detalle la distribución de datos, el MST resulta una alternativa adecuada ya que no necesita trabajar con un conjunto elevado de neuronas.

- La complejidad computacional del método es  $\mathcal{O}(NPT)$ , donde  $N$  es el número de patrones de entrada a proyectar,  $P$  es la cardinalidad del conjunto  $\Psi(x)$  y  $T$  depende del algoritmo de optimización (en este trabajo denota el número de veces que se calcula  $\varepsilon_x(p)$  antes de alcanzar un mínimo local). Si  $P$  es pequeño el método genera proyecciones en tiempos de ejecución aceptables independientemente del tamaño del MAO. Sin embargo, el método presenta mejores resultados cuando  $\Psi(x)$  contiene todos los pares de neuronas. En esos casos  $P = M(M - 1)/2$ , donde  $M$  es el número de neuronas del MAO. Por tanto, la complejidad computacional pasa a ser  $\mathcal{O}(NM^2T)$ , por lo que el tamaño del MAO es un factor clave de cara a la velocidad del método. En este sentido, el método resulta más apropiado cuando el tamaño del MAO es pequeño.

En cualquier caso, el método es capaz de generar imágenes adecuadas de las distribuciones cuando el MAO es grande, y el analista debe evaluar qué conjunto de pares de neuronas es más adecuado en cada situación, ya que también es posible generar descripciones de datos adecuadas utilizando conjuntos relativamente reducidos de pares de neuronas.

El método resulta ser robusto a la adaptación del MAO a la distribución en cuanto al error de cuantización y puede describir adecuadamente ciertos tipos de distribuciones difíciles de visualizar con la mayoría de métodos de visualización. Por otro lado, es general en el sentido de no estar limitado a la correspondencia discreta que genera un MAO, y tampoco está limitado a MAO bidimensionales.

Por otro lado, antes de su implementación se probó construir un método de interpolación basado en RNA supervisadas para construir la función  $f$ . Para ello, se entrenó un perceptrón multicapa que recibía como entrada un vector de referencia (en la fase de entrenamiento) o dato (para calcular su proyección, una

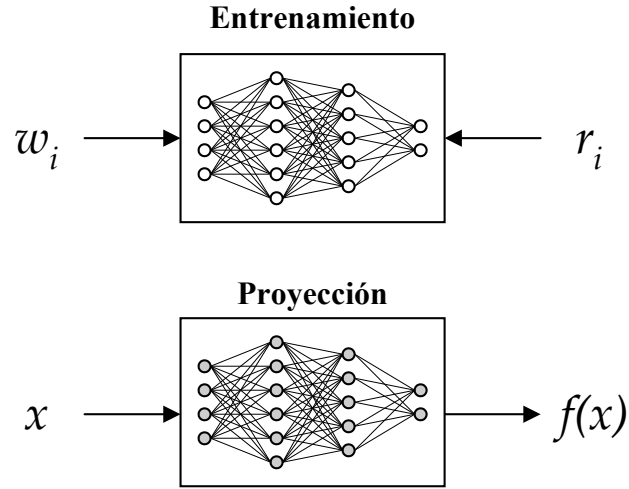


Figura 6.38: Proyección calculada a partir de un perceptrón multicapa.

vez entrenada la red), contenía varias capas ocultas, y la salida estaba formada por dos neuronas asociadas a dos coordenadas de un plano bidimensional. A esta red neuronal se le enseñaba a asociar los vectores de referencia  $w_i$  (entradas) del MAO con sus respectivas posiciones  $r_i$  (salidas) en la rejilla regular. Después de entrenar la red, ésta definía la proyección  $f$ , de manera que la proyección de un dato  $x$  correspondía a la salida de la red cuando se le introducía dicho dato en la entrada ( $f(x)$ ). Este proceso es ilustrado en la Figura 6.38. Sin embargo, a pesar de que este híbrido de un MAO con un perceptrón multicapa en ocasiones producía visualizaciones aceptables, fue descartado por varias razones. En general, no representaba las distribuciones de datos adecuadamente, y las proyecciones dependían en gran medida del perceptrón multicapa utilizado, de su arquitectura, de los diversos parámetros de entrenamiento, del orden de presentación de los datos de entrenamiento, etc., generando resultados muy distintos en cada ejecución. Por estos motivos, se decidió buscar una alternativa para prescindir del elemento de azar, que originó la creación del MST.

## 6.2. Superposición de Imágenes

Varios métodos de visualización de MAO producen imágenes monocromáticas que aportan información específica sobre los datos, las cuales suelen ser analizadas de forma individual, por ejemplo, matrices de distancias, histogramas de datos, planos de componentes, etc. En realidad, a la hora de examinar un MAO el

analista dispone de mucha información relacionada con cada neurona a través de estas imágenes. Aunque en la práctica cada método es empleado individualmente, desde un punto de vista el analista examina una distribución multidimensional a través de otros datos multidimensionales relativos a cada neurona del MAO (distancias entre los vectores de referencia, proporción de datos asociados a cada neurona, valor de los componentes de los vectores de referencia en cada neurona, etc.). Estos nuevos datos multidimensionales podrían ser analizados mediante cualquier técnica de análisis multidimensional (véase el Capítulo 2), sin embargo, en la práctica esto no es habitual dada la complejidad añadida que supone combinar tanta información, especialmente cuando cada método puede proporcionar información muy diferente.

No obstante, este apartado propone una estrategia para facilitar la labor del analista a la hora de combinar la información de un número limitado de métodos mediante una simple superposición de imágenes basada en un modelo aditivo de colores, siguiendo una estrategia similar a la empleada en la Figura 2.4. Definiendo las imágenes con diferentes colores primarios del modelo aditivo éstas pueden ser combinadas, por ejemplo, mediante una simple suma de sus componentes de color. Las imágenes resultantes son más completas y robustas, especialmente cuando se combinan imágenes que aportan información similar. Esta tesis estudia principalmente la combinación de la U-matrix (matriz de distancias) con histogramas de datos. La motivación detrás de esta idea reside en la posibilidad de mejorar el método U-matrix incorporando información acerca de las propias muestras de la distribución.

### 6.2.1. Descripción del problema

Los MAO deben ser analizados mediante varios métodos diferentes de visualización por varias razones:

- Siguiendo la filosofía del AED, el papel del analista es explorar los datos de tantas maneras como sea posible hasta que una “historia” plausible de los datos emerja [Beh97, Yu01].
- El MAO implementa una proyección en la que la dimensión de los datos es reducida, lo cual conlleva una pérdida de información. Por tanto, ningún método de visualización puede aportar una descripción completa de un conjunto de datos definido en un espacio de dimensión elevada.

- Puede que no exista un método óptimo para observar una determinada característica de los datos, por ejemplo, la estructura de los *clusters*.
- La utilidad de un método en concreto siempre depende de la distribución de datos a analizar.

Teniendo en cuenta estas recomendaciones y dado que el analista frecuentemente trabaja con imágenes que aportan información útil sobre la distribución de datos, éste debe considerar varios métodos de visualización de MAO simultáneamente. Esto puede ser llevado a cabo mostrando las diferentes imágenes generadas por los métodos en una configuración adecuada, por ejemplo, alineadas en una línea recta o siguiendo una configuración matricial. Desde un punto de vista práctico esto tiene dos inconvenientes dado el elevado tamaño de los MAO utilizados con el objetivo de visualizar y explorar datos. Nótese que al contar con un elevado número de neuronas (a veces mayor que el número de patrones de entrada de la distribución [Ult99]) el tamaño de las imágenes generadas a partir de los diferentes métodos de visualización también es elevado. El primer problema está relacionado con el número limitado de imágenes que pueden ser visualizadas simultáneamente en el monitor de un computador con una resolución apropiada. El segundo problema, de mayor importancia, concierne la habilidad o precisión humana para combinar la información de dos o más imágenes. Independientemente de la semejanza entre ellas, el analista tiene la difícil tarea de proyectar mentalmente una o varias imágenes sobre otra que está visualizando, lo cual puede provocar errores de precisión e interpretación, especialmente cuando el tamaño de las imágenes es elevado.

El problema abordado en este apartado consiste en la representación simultánea, en una sola imagen, de la información contenida en un conjunto limitado de imágenes (dos o tres) de igual dimensión. A la hora de combinar imágenes de varios métodos es posible distinguir dos situaciones:

- Las imágenes a combinar aportan información diferente. Por ejemplo, una U-matrix y un plano de componente, donde la primera contiene información sobre la estructura de los *clusters* y la segunda indica el valor de un componente de los vectores de referencia (imagen generalmente no utilizada para analizar la estructura de *clusters*, a pesar de que en ocasiones puede revelar bordes entre algunos *clusters*). En general, la interpretación de estas imágenes combinadas suele ser complicada.

- Las imágenes a combinar aportan información similar. En estos casos la combinación de imágenes permite observar diferencias entre métodos diseñados para producir resultados similares. De esta manera, es posible obtener imágenes más completas y robustas, ya que conclusiones similares pueden ser corroboradas y es posible detectar (y corregir) deficiencias asociadas a las representaciones de cada método.

Por tanto, el objetivo en este apartado consiste en diseñar un método que facilite la correcta interpretación de la imagen combinada, cuando las originales representan información similar, aunque no se aporte más información que la contenida en las imágenes originales.

### 6.2.2. Solución propuesta

El problema encuentra una solución en la elección de un método apropiado de visualización de elementos multidimensionales, por ejemplo, iconos circulares, caras de Chernoff, etc. (véase la Sección 2.5.2). Debido a que el conjunto de imágenes a combinar debe ser limitado y a que los métodos más populares de análisis visual de MAO generan imágenes monocromáticas, se propone una estrategia para combinarlas basada en un modelo aditivo de colores [Rus95, Cas96, Fai98]. En concreto, cada imagen monocromática a mezclar es definida mediante un color primario (escala de color) diferente del modelo aditivo. De esta manera, su superposición genera una imagen final en la que cada componente de color queda asociado a una y sólo una imagen.

Los experimentos están basados en el popular modelo RGB por comodidad a la hora de implementar la combinación (a pesar de sus conocidas limitaciones, véanse [Hea96, Fai98]), por lo que el máximo número de imágenes a combinar es tres. Según este modelo, fundamentado en la teoría tricromática (basada en los trabajos de Maxwell, Young y Helmholtz), cada pixel de una imagen es definido mediante tres valores de intensidad luminosa (R,G,B), los cuales están asociados a tres regiones del espectro electromagnético visible (rojo, verde y azul). Con este modelo cada imagen es definida según una escala diferente de rojo, verde o azul, es decir, sólo un componente de color de cada imagen puede tomar un valor distinto de cero. De esta manera, la superposición de las imágenes puede ser realizada mediante una simple suma (u otra operación básica como una OR lógica) de sus componentes de color.



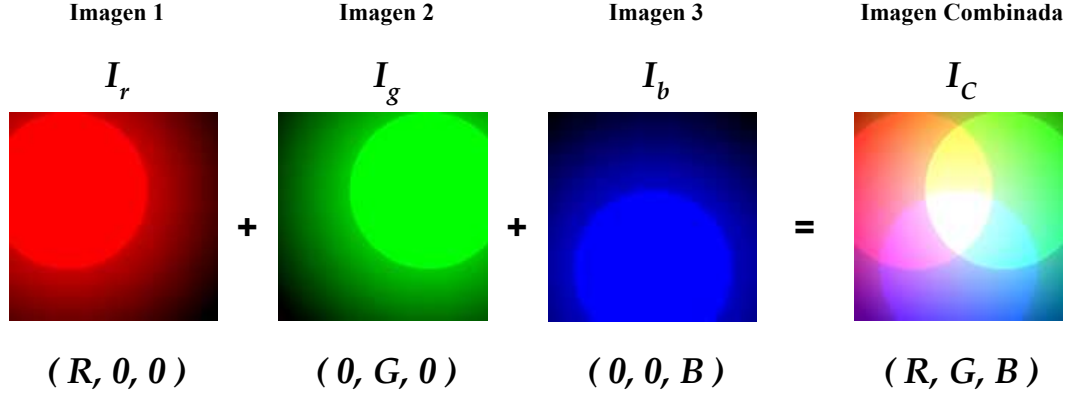


Figura 6.39: Superposición de imágenes monocromáticas según el modelo de color RGB.

Formalmente, sean  $\mathcal{I}_r$ ,  $\mathcal{I}_g$  y  $\mathcal{I}_b$  tres imágenes monocromáticas de igual dimensión, definidas según los componentes rojo, verde y azul del modelo RGB, respectivamente. Cada pixel  $x$  de las imágenes queda representado por un vector de tres componentes de color:

$$\mathcal{I}_r(x) = (i_{r,x}, 0, 0) \quad \mathcal{I}_g(x) = (0, i_{g,x}, 0) \quad \mathcal{I}_b(x) = (0, 0, i_{b,x})$$

El color de los pixeles de la imagen combinada  $\mathcal{I}_C$  queda definido por:

$$\mathcal{I}_C(x) = \mathcal{I}_r(x) + \mathcal{I}_g(x) + \mathcal{I}_b(x) = (i_{r,x}, i_{g,x}, i_{b,x}) \quad (6.13)$$

El proceso es ilustrado en la Figura 6.39, donde tres imágenes monocromáticas son sumadas para generar la correspondiente imagen combinada.

El número de posibles colores que pueden aparecer en la imagen final crece exponencialmente con respecto al número de imágenes a combinar. Por ejemplo, si cada componente de color puede tomar  $p$  valores diferentes el número de posibles colores resulta ser  $p^n$ , donde  $n$  es el número de imágenes a combinar. En esta tesis todas las imágenes generadas por los nuevos métodos están basadas en el modelo RGB, donde cada componente de color queda definido por un byte. De esta manera, en teoría es posible generar hasta  $256^3 = 16777216$  colores diferentes al mezclar tres imágenes, aunque en la práctica el sistema visual humano no es capaz de diferenciar tantos colores. En los experimentos la mezcla de tres imágenes puede derivar problemas de interpretación por el elevado número de colores que pueden aparecer en la imagen final, especialmente si las imágenes aportan información distinta. Aunque el ser humano es capaz de distinguir millones de

colores diferentes, encontrar dos puntos en una imagen con exactamente el mismo color requiere un esfuerzo considerable cuando ésta contiene un número elevado de diferentes colores [Ves02]. Según Healey [Hea96], el número máximo de colores que pueden ser separados entre sí de manera preatenta es de unos siete. Por otro lado, la descomposición de la imagen combinada en sus tres componentes primitivos no es una tarea fácil, es decir, suele ser difícil precisar la cantidad de rojo, verde y azul asociada a un determinado color.

Por el contrario, la combinación de dos imágenes suele producir resultados intuitivos, al menos desde un punto de vista subjetivo. Al reducirse la complejidad la cantidad de colores a interpretar suele ser abordable y resulta más sencillo precisar las cantidades de los dos colores mezclados en un pixel o región de la imagen final. De todas formas, la combinación de tres imágenes puede ser útil desde un punto de vista exploratorio, dado que una tercera imagen aporta más información y precisión. Por ejemplo, el analista puede estar interesado en localizar zonas específicas del MAO con un determinado color uniforme, o puede localizar zonas donde no esperaba encontrar determinados colores.

#### 6.2.2.1. U-matrix + histograma de datos

En realidad, el propósito de la combinación de imágenes es mejorar el método U-matrix (o las matrices de distancias), que es el más popular para analizar la estructura de *clusters*, incorporando información sobre las propias muestras de la distribución mediante histogramas de datos. Ambos métodos muestran la estructura de *clusters*, por lo que suelen generar imágenes parecidas, aportando información similar, ya que los MAO aproximan la función de densidad de probabilidad de la distribución del conjunto de datos. De esta manera, los vectores de referencia tienden a adaptarse a zonas de alta densidad de datos (*clusters*), donde las distancias entre ellos son relativamente pequeñas.

Sin embargo, aunque la U-matrix es capaz de proporcionar visiones útiles de una distribución de datos, al no considerar las muestras de la distribución puede presentar deficiencias a la hora de representar ciertas distribuciones [RGD<sup>+</sup>02]. Destacan las dos siguientes situaciones problemáticas:

- Los vectores de referencia pueden hallarse relativamente cerca de sus vecinos sin apenas representar datos, especialmente cuando el tamaño del MAO es elevado (como suele ocurrir cuando el objetivo es visualizar datos).

- Los vectores de referencia pueden hallarse relativamente lejos de sus vecinos representando a una cantidad elevada de datos, por lo que puede ser difícil apreciar las clases o *clusters* asociados a ellos.

Por tanto, un análisis de las distancias entre vectores de referencia debe ser complementado con un análisis de la proporción de datos asociados a cada vector de referencia (información que aporta el histograma de datos por definición).

La Figura 6.40 ilustra dos ejemplos de los problemas asociados a la U-matrix. Aunque dichos problemas pueden ser resueltos empleando MAO de mayor tamaño, por claridad los ejemplos emplean MAO  $(5 \times 5)$ -dimensionales, donde el número de neuronas parece suficiente como para modelar las distribuciones adecuadamente (obsérvese que un algoritmo de clasificación basado en la distancia Euclídea presentaría una tasa de acierto cercana al 100 %). En ambos casos queda patente la necesidad de complementar la información acerca de las distancias entre vectores de referencia con información sobre los propios datos de la distribución.

En (a) se muestra un MAO adaptado a la distribución formada por cuatro *clusters* de igual tamaño pero distinta forma, donde aparecen cuatro neuronas interpolantes cuyos vectores de referencia no se han adaptado a ningún *cluster* específico: dos entre los *clusters* de la izquierda y otros dos entre los *clusters* inferiores. Éstas no representan a ningún dato de entrada, a pesar de que la distancia entre sus vectores de referencia y los de sus vecinas es relativamente pequeña. De esta manera, aparecen varias deficiencias en la U-matrix (b): a pesar de que sólo dos vectores de referencia se han adaptado al *cluster* superior-izquierdo, la imagen de la U-matrix parece indicar que las cuatro neuronas de la esquina superior-izquierda del MAO se han adaptado al *cluster*. Por otro lado, no aparecen con claridad los bordes entre los *clusters* inferiores. Estos problemas pueden ser corregidos incorporando información del histograma de datos (c), el cual refleja mejor las características de la distribución: cuatro *clusters* disjuntos, densidad relativa de patrones en cada *cluster* y forma (nótese que todos los *clusters* contienen el mismo número de patrones por lo que la intensidad luminosa media de cada *cluster* resulta ser directamente proporcional al área que ocupa en el mapa).

El segundo ejemplo de la figura ilustra un MAO adaptado a la distribución formada por un círculo y un anillo (d). El problema en este caso reside en las distancias relativamente elevadas entre los vectores de referencia adaptados a

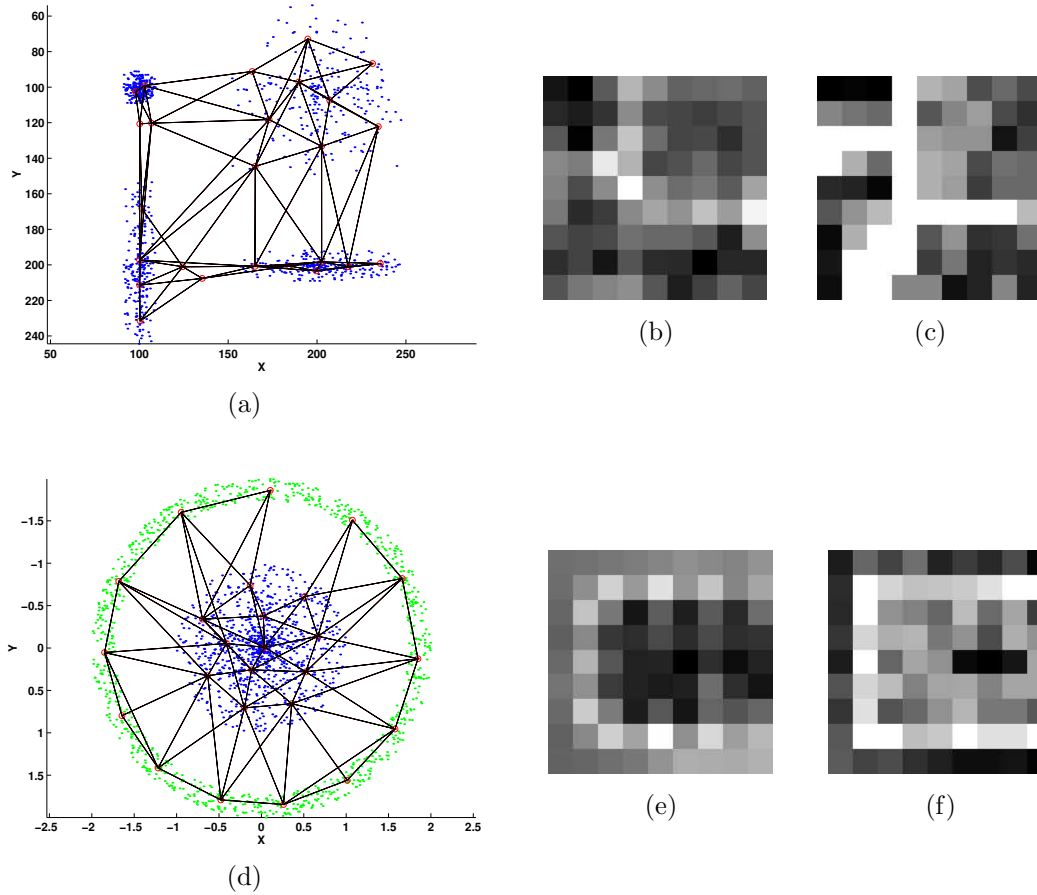
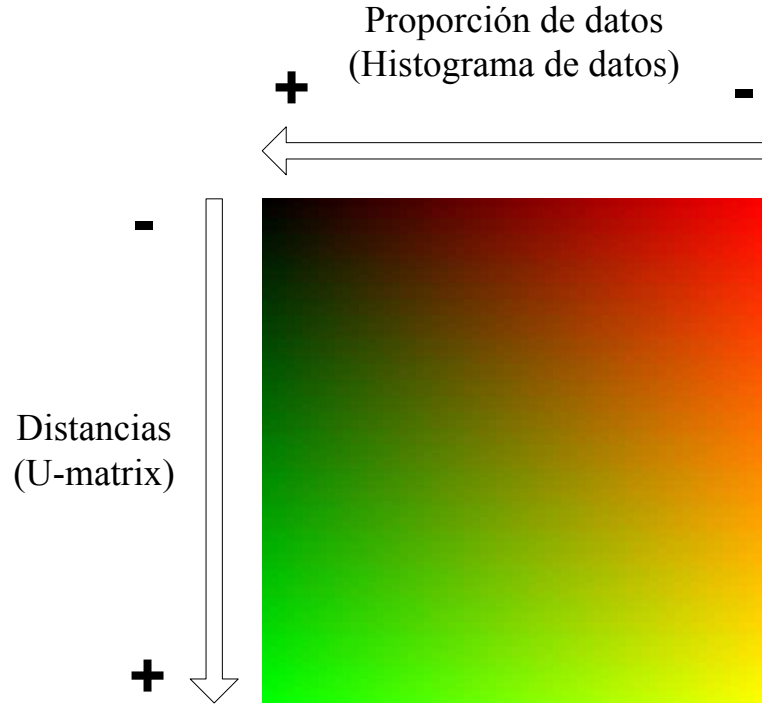


Figura 6.40: Problemas del método U-matrix que pueden ser resueltos incorporando información del histograma de datos, que en estos casos representa mejor la distribución. Se ilustra un MAO (5 × 5)-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño pero distinta forma (a), y otro a la distribución formada por un círculo y un anillo (d). Las imágenes (b) y (c) muestran la U-matrix y el histograma de datos, respectivamente, para el primer MAO, y (e) y (f) para el segundo. En la primera U-matrix no es posible apreciar las cuatro neuronas interpolantes por lo que no se distinguen bien los bordes entre los *clusters*. En la segunda, el anillo exterior no es ilustrado con claridad debido a la mayor distancia entre los vectores de referencia.

Figura 6.41: Plano  $B = 0$  del cubo RGB.

la clase formada por el anillo exterior. La U-matrix (e), que sirve principalmente para detectar *clusters*, lógicamente tiene dificultades para describir el anillo por su forma particular. En la imagen es posible apreciar un *cluster* central claramente, pero no es trivial determinar la existencia de un anillo exterior. Sin embargo, el histograma de datos (f), ilustra claramente una clase alrededor del *cluster* central. En cuanto a la implementación de los métodos, en ambos ejemplos se ha ecualizado el histograma de las imágenes, y el histograma de datos ha sido generado según la variante descrita en la Tabla 5.3.

En los experimentos la U-matrix y el histograma de datos han sido definidos mediante escalas de verdes y rojos, respectivamente, mientras que el componente azul permanece constante e igual a cero. No se aprecia una diferencia significativa en los resultados al utilizar otras combinaciones de colores (rojo y azul, o verde y azul), si bien el rojo y verde son dos colores opuestos de la teoría de colores oponentes de Ewald Hering (véase [Fai98]), lo cual podría tener una ligera repercusión a la hora de interpretar los diferentes colores que aparecen en la imagen final (véanse las líneas futuras de esta tesis). Las imágenes aprovechan todo el rango dinámico de intensidad luminosa (véase (5.1)) y los histogramas













Superposición	U-matrix	Histograma de datos
Color Oscuro 	distancias cortas 	densidad elevada 
Verde 	distancias largas 	densidad elevada 
Rojo 	distancias cortas 	densidad pequeña 
Amarillo 	distancias largas 	densidad pequeña 

Tabla 6.4: Simplificación de las familias de colores a detectar al superponer una U-matrix (definida mediante una escala de verdes) y un histograma de datos (definido mediante una escala de rojos).

de las imágenes originales generalmente han sido ecualizados antes de realizar la superposición. De esta manera, en la imagen final podría aparecer cualquier combinación de rojo y verde, es decir, cualquier color representado en la Figura 6.41, que ilustra el plano  $B = 0$  del cubo RGB.

Dado que el objetivo de la combinación de la U-matrix con el histograma de datos es realizar un análisis mejorado de la estructura de los *clusters* de la distribución, en la práctica el analista debe buscar los siguientes grandes grupos o familias de colores:

- Tonos oscuros. Indican claramente la existencia de una zona de alta densidad de un *cluster*, ya que las distancias entre los vectores de referencia de neuronas vecinas son cortas (valor oscuro en la U-matrix) y la proporción de patrones asociados a cada neurona es elevada (valor oscuro en el histograma de datos).
- Verdes. En este caso las distancias entre los vectores de referencia son mayores a pesar de que éstos siguen representando una cantidad elevada de datos. Suelen indicar *clusters* dispersos cuando el área verde es extensa, *clusters* aislados si ésta es pequeña, fronteras de *clusters* o incluso bordes entre *clusters*.
- Rojos. Indican vectores de referencia cercanos a sus vecinos pero con poca densidad de datos en torno a ellos. Pueden revelar defectos en la formación

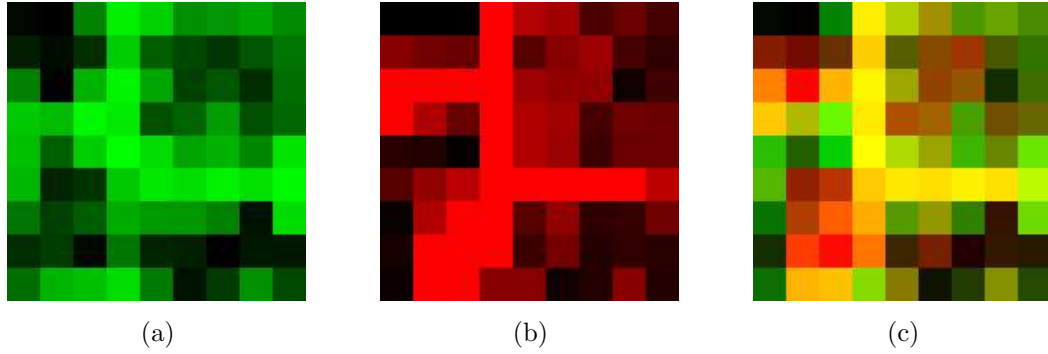


Figura 6.42: Superposición de imágenes para un MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y distinta forma. U-matrix (a), histograma de datos (b) y superposición (c).

del MAO, neuronas interpolantes, o simplemente zonas de clases con poca densidad relativa.

- Amarillos. Es el caso opuesto al de los colores oscuros, por lo que representan neuronas interpolantes y claros bordes entre *clusters*.

La Tabla 6.4 resume esta clasificación. Las zonas oscuras y amarillas corresponden a áreas del MAO donde la U-matrix y el histograma de datos presentan el mismo resultado, mientras que las verdes y rojas corresponden a áreas donde éste difiere. De esta manera, la imagen final es más robusta que las dos originales ya que permite corroborar sus resultados (colores oscuros y amarillos) y detectar sus diferencias (verdes y rojos). En comparación con la U-matrix, proporciona un análisis más fiable de los *clusters* y de los bordes entre éstos al incorporar información sobre la densidad de datos.

### 6.2.3. Resultados experimentales

Este apartado muestra varios ejemplos donde es posible apreciar como la U-matrix y el histograma de datos pueden complementarse, independientemente del tamaño del MAO utilizado.

#### 6.2.3.1. MAO pequeños

La Figura 6.42 muestra la superposición de las imágenes descritas en la Figura 6.40. La U-matrix (a) en tonos de verde y el histograma de datos (b) en tonos de rojo son superpuestos en (c), para el MAO  $(5 \times 5)$ -dimensional adaptado a la

distribución formada por cuatro *clusters* de igual tamaño y distinta forma. En este caso, el histograma de datos indica la existencia de cuatro *clusters* disjuntos y la densidad de datos en cada uno de ellos, mientras que la U-matrix aporta información sobre la distancia relativa entre cada *cluster*. Por tanto, los bordes entre los *clusters* se localizan donde aparecen zonas rojas, naranjas o amarillas, es decir, donde las neuronas apenas representan datos. Como era de esperar, el *cluster* superior-derecho queda claramente separado del resto (algo que no es posible observar en el histograma de datos), tal como indica el borde amarillo, ya que las distancias entre los vectores de referencia son más elevadas en esa zona del mapa. Obsérvese que dichas zonas amarillas no corresponden a neuronas interpolantes ya que no están ubicadas directamente sobre neuronas en el mapa. El resto de bordes no son tan claros debido a que la U-matrix no los ilustra adecuadamente. Sin embargo, con la información del histograma de datos el analista puede localizar las neuronas interpolantes, las cuales van a formar parte del resto de bordes (zonas naranjas y rojas). Por otro lado, en la superposición es posible observar como el *cluster* superior-izquierdo está formado únicamente por dos neuronas.

La Figura 6.43 muestra la U-matrix (a) en tonos de verde, el histograma de datos (b) en tonos de rojo, y la superposición resultante del método (c), para el MAO ( $5 \times 5$ )-dimensional adaptado a la distribución formada por un círculo y un anillo. La imagen combinada muestra claramente las dos clases de la distribución y el borde entre éstas. El anillo exterior queda representado en un tono verde que indica una clase compuesta por una elevada cantidad de datos, pero más dispersa que la del círculo (lo cual no queda claro en el histograma de datos). En la zona del MAO asociada al círculo aparece una zona muy oscura en el centro mientras que el resto aparece con un color rojizo. Esto es debido a la distribución no-uniforme del círculo, donde existe una mayor cantidad de datos en su centro. Como el anillo y el círculo poseen la misma cantidad de datos, los vectores de referencia asociados a la zona exterior del círculo representan menos datos a pesar de encontrarse cerca unos de otros, por lo que quedan representados mediante tonos rojos. Finalmente, aunque no existen neuronas interpolantes, el borde amarillo entre las dos clases también es claramente visible.

En ocasiones el analista suele estar interesado principalmente en hallar los *clusters* y los bordes entre éstos. La imagen (d) muestra una combinación diferente de la U-matrix y el histograma de datos, que generalmente produce buenos



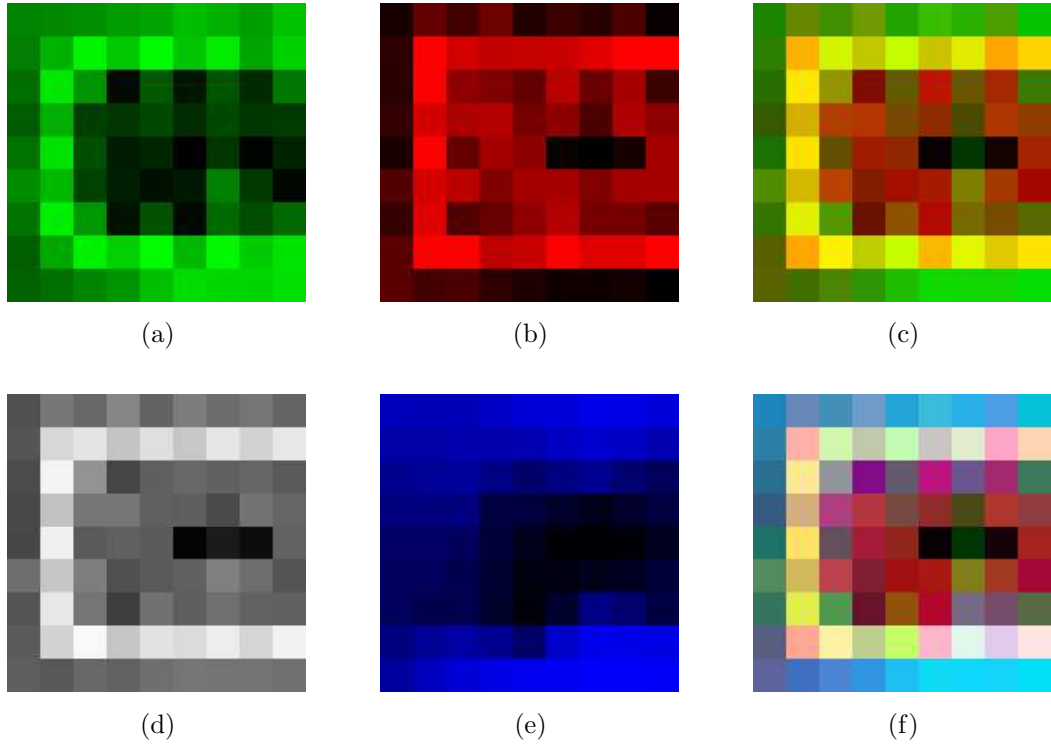


Figura 6.43: Superposición de imágenes para un MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por un círculo y un anillo. U-matrix (a), histograma de datos (b) y superposición (c). La imagen (d) muestra el promedio de las imágenes (a) y (b) definidas mediante tonos de gris. El error de cuantización de cada neurona es representado en (e) mediante tonos de azul. La superposición de (a), (b) y (e) es mostrada en (f).

resultados para detectar estas estructuras. El proceso consiste en calcular la media entre las dos imágenes representadas mediante niveles de gris. Sea  $\mathcal{I}_U(x)$  el nivel de gris del pixel  $x$  de la U-matrix y  $\mathcal{I}_H(x)$  el del histograma de datos, el nivel de gris de la imagen combinada  $\mathcal{I}_C(x)$  para ese pixel queda definido mediante:

$$\mathcal{I}_C(x) = \frac{\mathcal{I}_U(x) + \mathcal{I}_H(x)}{2} \quad (6.14)$$

De esta manera, las zonas donde las dos imágenes son similares quedan claramente representadas mediante un color oscuro o claro, que van a corresponder a los *clusters* y a los bordes, respectivamente. Esto es equivalente a hallar las zonas oscuras y amarillas de la superposición coloreada. Aunque esta nueva superposición no es tan completa como la primera (no describe la proporción de gris asociada a cada imagen original) resulta un método eficaz y sencillo de interpretar.

Una tercera imagen monocromática puede ser combinada junto con la U-matrix y el histograma de datos utilizando el último componente disponible del modelo RGB (azul). Sin embargo, el analista debe escoger con mucho cuidado la información a añadir para no complicar la interpretación de la nueva imagen. Una imagen que en ocasiones puede revelar información acerca de la estructura de los *clusters* es una representación del error de cuantización asociado a cada vector de referencia. Si éste es bajo, es razonable pensar que el vector de referencia puede hallarse en una zona de alta densidad de datos y por tanto cerca de sus vecinos. Por el contrario, un vector de referencia que represente a pocos patrones generalmente se hallará lejos de éstos (lo que provocará un elevado error de cuantización) y de sus vecinos. La imagen (e) muestra una imagen suavizada del error de cuantización asociado a cada vector de referencia en tonos de azul, donde la intensidad luminosa es directamente proporcional a éste, y máxima cuando un vector de referencia no representa a ningún patrón. Para el MAO y la distribución utilizados la imagen es similar a la U-matrix. Finalmente, la imagen (f) representa la superposición de (a), (b) y (e). El resultado es similar al de (c), aunque queda claro que la imagen es más difícil de interpretar debido a la complejidad añadida y a los múltiples colores que aparecen en ella.

### 6.2.3.2. Círculo y anillo

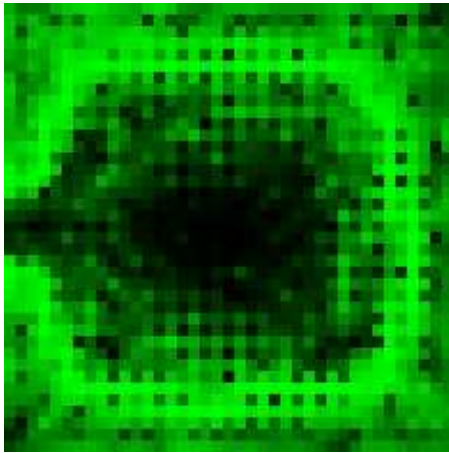
En los ejemplos anteriores los MAO eran pequeños. Aunque con ellos es posible determinar ciertas propiedades de las distribuciones, en la práctica suelen ser de

mayor tamaño cuando el objetivo es visualizar datos. Los siguientes experimentos utilizan MAO de tamaño medio y elevado.

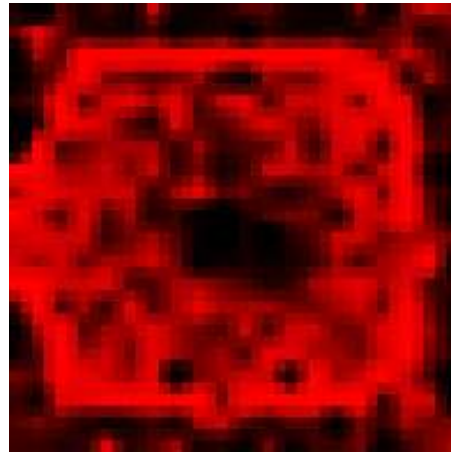
La Figura 6.44 muestra la U-matrix (a) en tonos de verde, el histograma de datos (b) en tonos de rojo, y la superposición resultante del método (c), para el MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un círculo y un anillo (véase la Figura 6.33). La única diferencia con respecto al ejemplo anterior es el tamaño superior del MAO. Esto permite representar mejor la distribución de datos (parece más clara la forma circular de la clase interior, rodeada casi completamente por la clase exterior). En el ejemplo se sigue apreciando la distribución no uniforme del círculo, donde existe una mayor concentración de datos en su centro (oscuro en el interior y progresivamente más rojizo hacia su exterior). También aparece claramente el borde amarillo entre las dos clases y el anillo exterior verde que indica una clase más dispersa.

### 6.2.3.3. Cuatro *clusters* disjuntos de diferente tamaño e igual forma

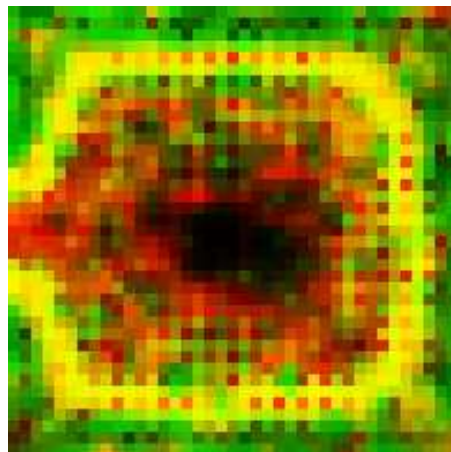
La Figura 6.45 muestra la U-matrix en tonos de verde antes (a) y después (b) de ecualizar el histograma de la imagen, el histograma de datos (c) en tonos de rojo, y la superposición resultante del método (d), para el MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma (véase la Figura 6.28). Este ejemplo resulta interesante por varias razones. En primer lugar, como los vectores de referencia de los *clusters* laterales (centrados en  $X1 = -10$  y en  $X1 = 10$ ) se hallan muy cerca entre sí, un primer análisis de la U-matrix original no permite detectar los bordes entre dichos *clusters*, por lo que parece indicar la existencia de únicamente dos clases. Este problema puede ser mitigado realizando una ecualización del histograma de la imagen, donde resulta más fácil detectar cuatro clases. Sin embargo, la imagen resultante es más “ruidosa”. Por otro lado, la imagen del histograma de datos (que también ha sido ecualizada) está muy relacionada con la imagen ecualizada de la U-matrix y también revela los bordes estrechos entre los *clusters* laterales. Curiosamente, en las zonas que corresponden a los *clusters*, la U-matrix parece la imagen negativa del histograma de datos. Es decir, ambas imágenes parecen representar información opuesta dentro de los *clusters* (las zonas oscuras y verdes de la U-matrix aparecen rojas y oscuras, respectivamente, en el histograma de datos). Aunque aparentemente contradictorio (las zonas oscuras deberían coincidir), debido a la distribución de datos uniforme dentro de los *clusters*, si



(a)



(b)



(c)

Figura 6.44: Superposición de imágenes para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un círculo y un anillo. U-matrix (a), histograma de datos (b) y superposición (c).

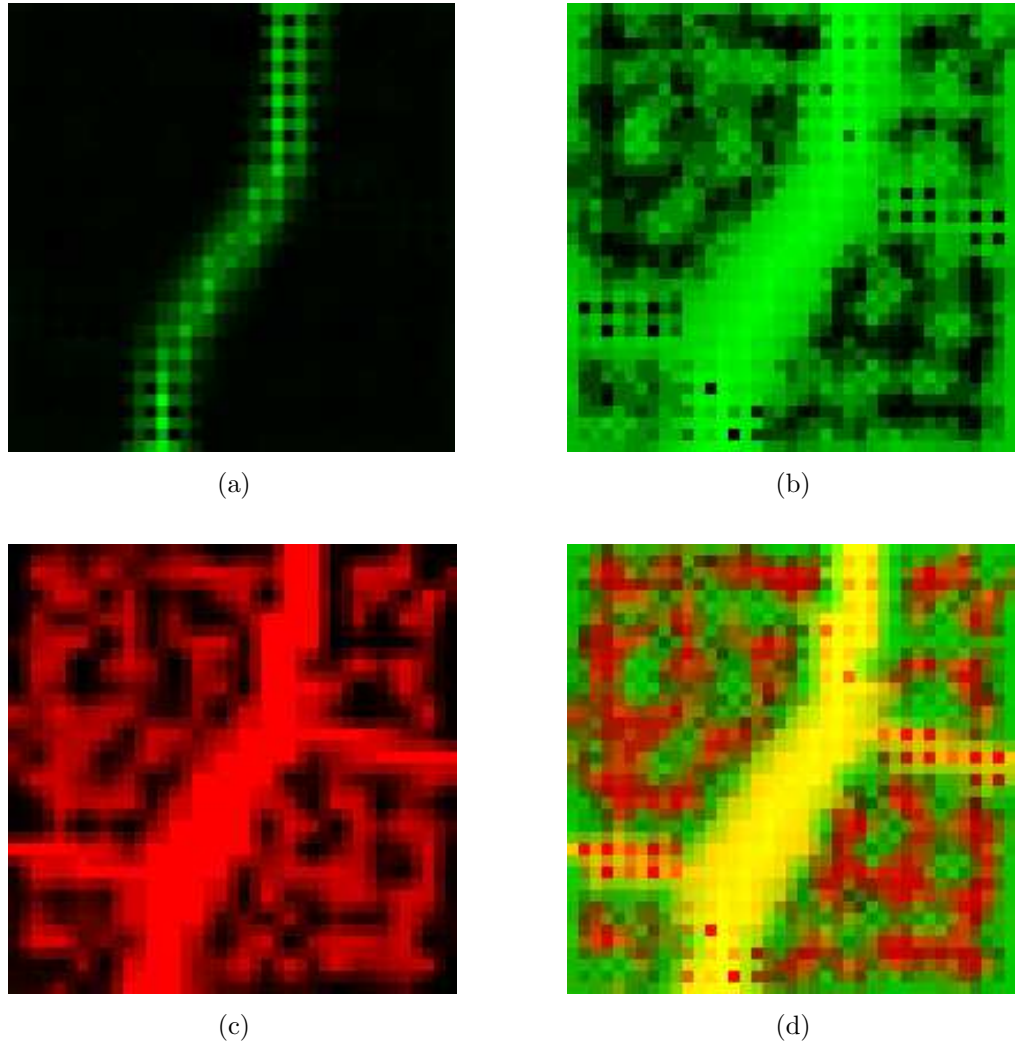


Figura 6.45: Superposición de imágenes para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma. U-matrix (a), U-matrix después de ecualizar el histograma de la imagen (b), histograma de datos (c), y superposición de (b) y (c) en (d).

un vector de referencia se halla cerca de sus vecinos su celda de Voronoi va a ser relativamente más pequeña y por tanto va a representar a una cantidad menor de datos. Por el contrario, si se halla lejos va a representar a más datos. En general, cuando una región de un MAO se ajusta a una distribución uniforme los vectores de referencia tienden a distribuirse también de manera uniforme, donde las distancias entre éstos y sus vecinos resultan similares. De esta manera, la U-matrix y el histograma de datos suelen representar esas regiones mediante un color prácticamente uniforme, donde las diferencias entre los colores no son significativas. Sin embargo, al realizar la ecualización del histograma puede suceder la situación presentada en este ejemplo, donde esas pequeñas diferencias son amplificadas y la U-matrix parece indicar información opuesta al histograma de datos. El resultado es una superposición que no muestra regiones claramente oscuras, donde los *clusters* aparecen compuestos de una mezcla de tonos rojos y verdes. Finalmente, los bordes amarillos son claramente visibles (los laterales contienen neuronas interpolantes cuyos vectores de referencia se hallan cerca de los de sus vecinas, por lo que son distinguibles por su color rojo intenso).

#### 6.2.3.4. Ocho *clusters* disjuntos con forma hipercúbica

La Figura 6.46 muestra la U-matrix (a) en tonos de verde, el histograma de datos (b) en tonos de rojo, y la superposición resultante del método (c), para el MAO (20 × 20)-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos (véase la Figura 6.27). Este experimento es un ejemplo sencillo, donde las imágenes a superponer son muy similares, y donde es posible apreciar una vez más como la U-matrix puede ser complementada con el histograma de datos. Información acerca de las distancias entre *clusters* (sólo visible en la U-matrix) es combinada con información sobre la densidad de datos en cada *cluster* (más fácil de ver en el histograma de datos). De esta manera, en la imagen final es posible observar los claros bordes amarillos (mejor definidos donde aparece un amarillo más puro o menos “borroso”), las fronteras de los *clusters* en verde y la densidad de datos dentro de cada *cluster* en función de su tono rojizo. En este ejemplo las clases a las que se adaptan los vectores de referencia siguen distribuciones uniformes. Sin embargo, el efecto debido a la ecualización no aparece tan claramente como en el ejemplo anterior, especialmente en el histograma de datos, por lo que existe menos varianza en los colores que aparecen en los *clusters*.

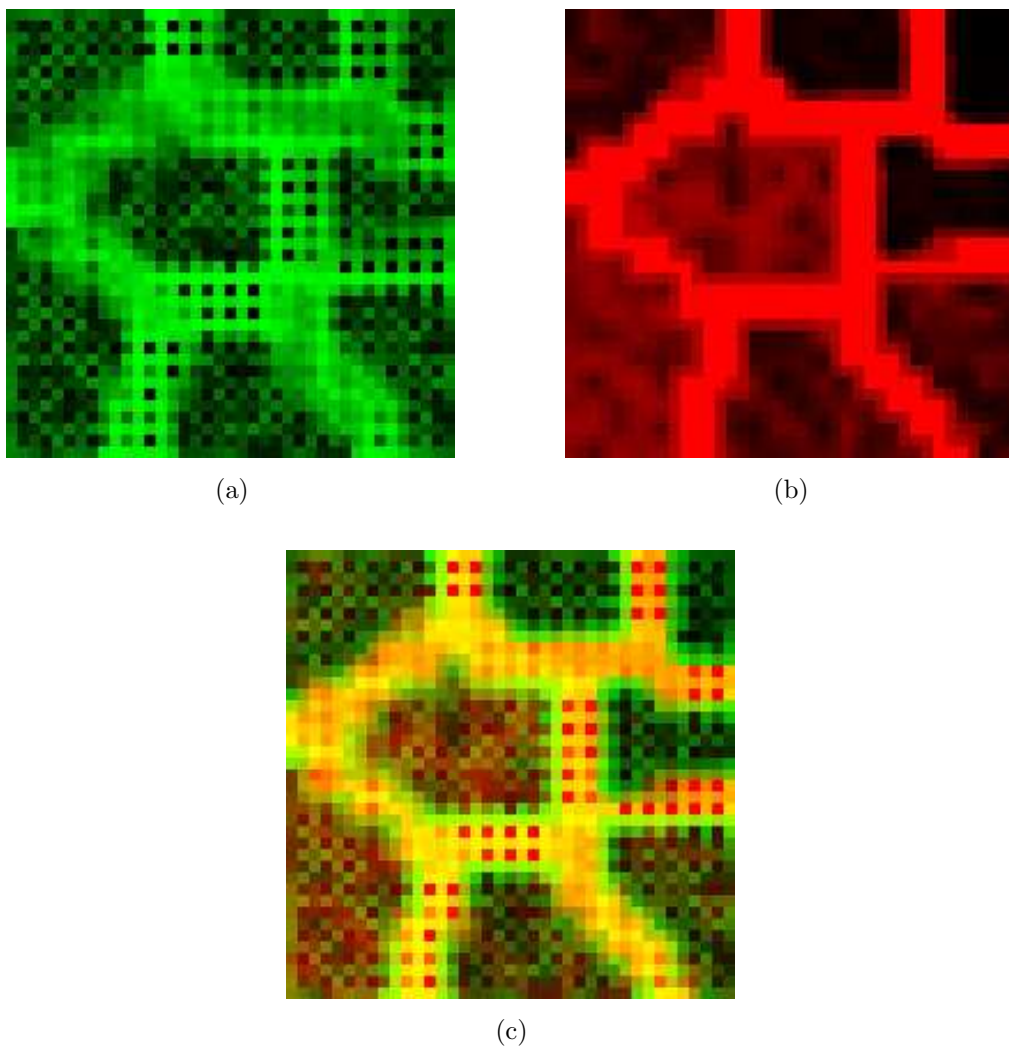


Figura 6.46: Superposición de imágenes para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos. U-matrix (a), histograma de datos (b) y superposición (c).

### 6.2.3.5. Dos eslabones encadenados

La Figura 6.47 muestra la matriz de distancias antes (a) y después (b) de ecualizar su histograma en tonos de verde, el histograma de datos ecualizado (c) en tonos de rojo, y la superposición resultante de las imágenes ecualizadas (d), para un MAO ( $100 \times 100$ )-dimensional adaptado a la distribución formada por dos anillos encadenados. Debido al elevado número de neuronas del MAO tanto las clases como los bordes entre éstas son representados mediante muchas neuronas, lo cual genera anchos bordes entre las clases con multitud de neuronas interpolantes y extensas zonas representando *clusters*. De esta manera, a pesar de que la dimensión intrínseca de los datos es uno, la matriz de distancias representa los dos anillos mediante regiones oscuras anchas (especialmente en la imagen original sin ecualizar). Sin embargo, no todas las neuronas de estas zonas oscuras representan datos. De hecho, la mayor parte de las neuronas en estas zonas son neuronas interpolantes. Esto queda claro en el histograma de datos, el cual muestra con más precisión las neuronas que han sido ajustadas a la distribución, apareciendo una serie de curvas parecidas a las de Peano (véase la Sección 3.2.5.1). Por otro lado, las discontinuidades de estas curvas dificultan la interpretación del número exacto de clases que forman la distribución, que por el contrario, es más fácil de determinar utilizando la matriz de distancias. Finalmente, el lector puede comprobar como, dado el tamaño elevado del MAO, no es sencillo determinar sobre la matriz de distancias qué neuronas representan datos, incluso visualizando el histograma de datos simultáneamente a un lado. La superposición de imágenes constituye una solución a estos problemas al combinar la información de los dos métodos en una sola imagen.

### 6.2.3.6. Circunferencia, segmento, cuadrado y ortoedro

La Figura 6.48 muestra la U-matrix en tonos de verde antes (a) y después (b) de ecualizar el histograma de la imagen, el histograma de datos (c) en tonos de rojo, y la superposición resultante del método (d), para el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro (véase la Figura 6.35). En este caso existe una clara dirección de mayor varianza en los datos (el eje  $X$ ). Por tanto, dada su forma cuadrada, el MAO se adapta a la distribución (adecuadamente) de tal forma que ésta queda representada a lo largo de una de las diagonales (la secundaria de  $45^\circ$ ). Esto provoca que las neuronas de las esquinas superior-izquierda e



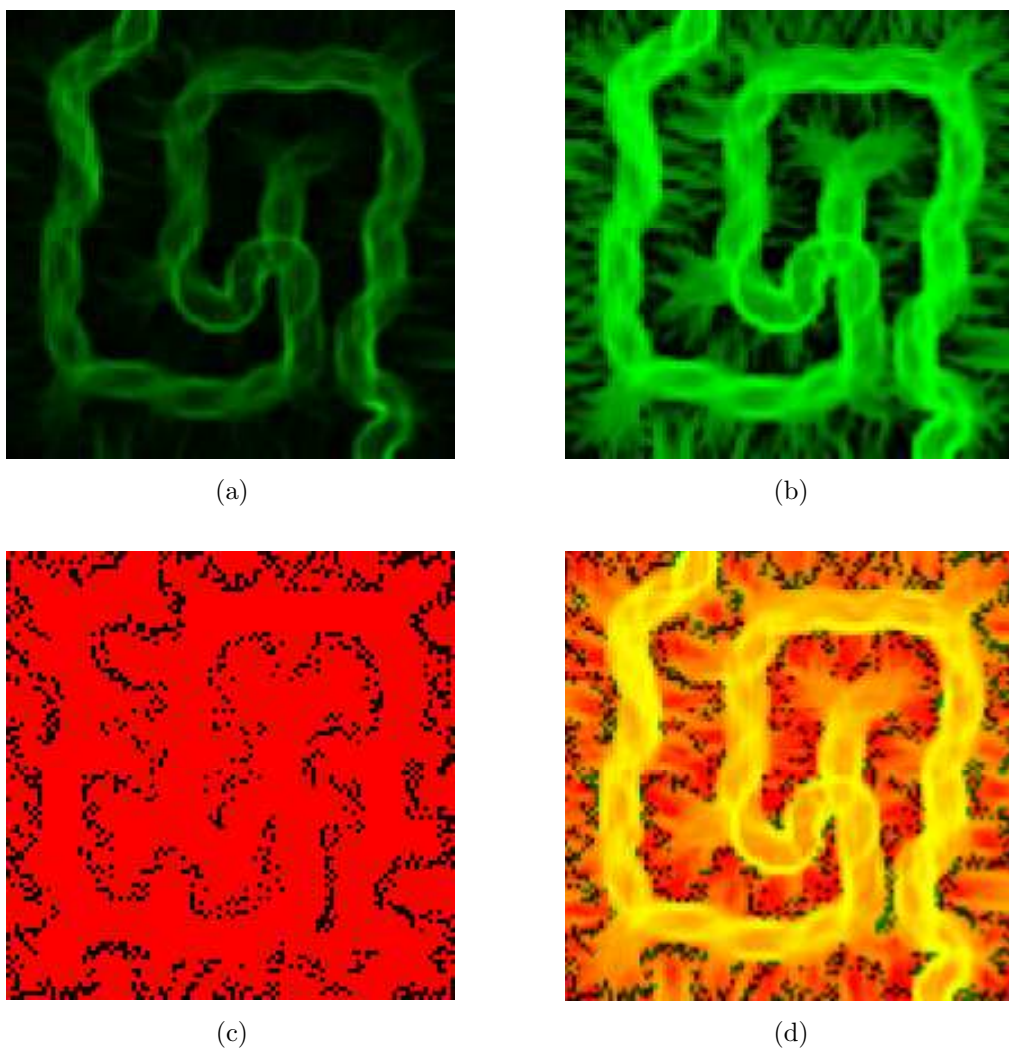


Figura 6.47: Superposición de imágenes para un MAO ( $100 \times 100$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados. Matriz de distancias antes (a) y después (b) de ecualizar su histograma, histograma de datos ecualizado (c), y superposición de (b) y (c) en (d).

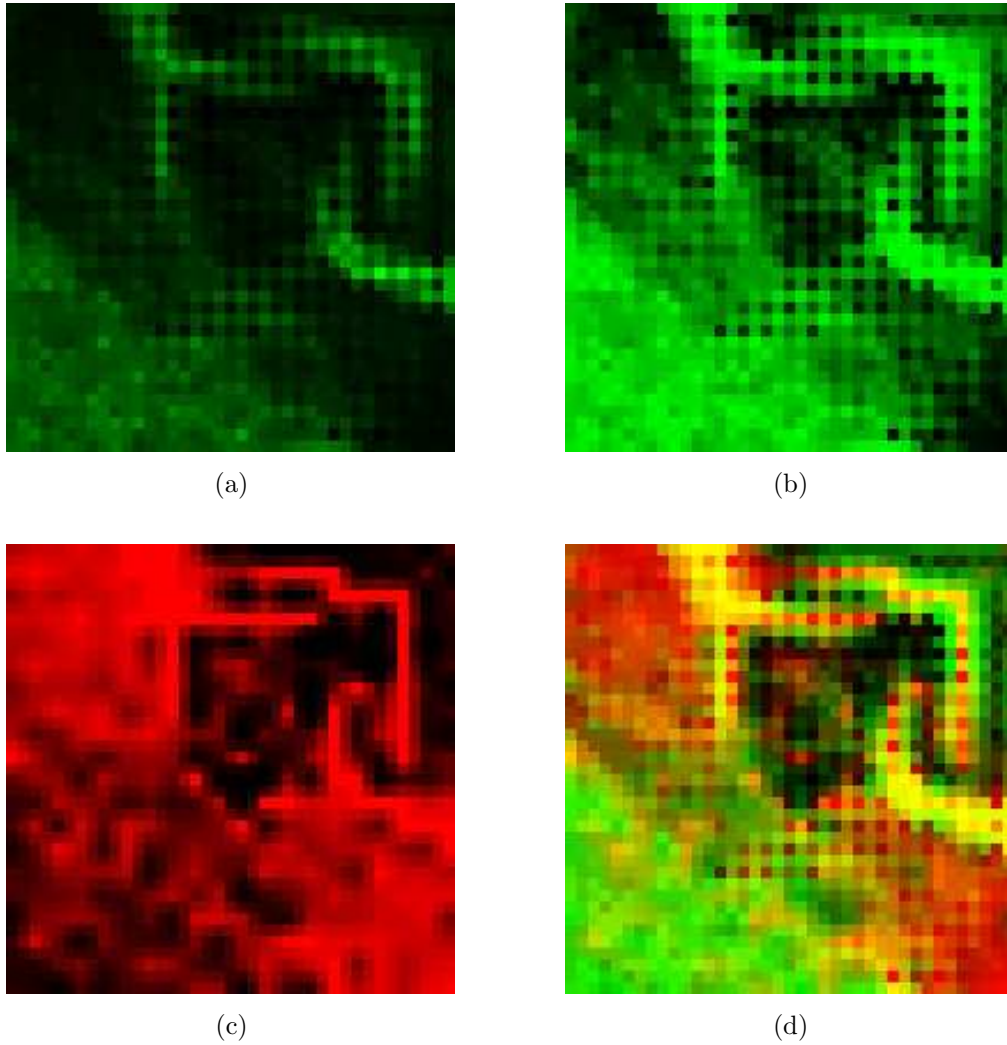


Figura 6.48: Superposición de imágenes para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro. U-matrix (a), U-matrix después de ecualizar el histograma de la imagen (b), histograma de datos (c) y superposición de (b) y (c) en (d).

inferior-derecha apenas representen datos, a pesar de que sus vectores de referencia se encuentran muy cerca de los de sus vecinas. Este problema naturalmente aparece en la U-matrix, donde parece que existen dos grandes *clusters* en dichas esquinas, cuando en realidad esas zonas están compuestas de neuronas interpolantes. En cualquier caso, en la U-matrix es posible visualizar la existencia de las cuatro clases que componen la distribución. En la esquina superior-derecha se encuentra la circunferencia, donde es posible distinguir una forma cíclica. El segmento también es visible, a pesar de que aparece como un *cluster* compacto donde resulta imposible intuir su dimensión intrínseca. Para ello es necesario recurrir a otros métodos de visualización como los tres propuestos en las Secciones 6.1, 6.3 y 6.4. El ortoedro queda distorsionado y es fácil reconocerlo en la esquina inferior-izquierda, donde naturalmente existe mayor distancia entre los vectores de referencia que han sido adaptados a él, por lo que la U-matrix lo muestra mediante un color verde intenso. Finalmente, la superficie cuadrada, algo más oscura, aparece entre el ortoedro y el segmento. Ésta resulta más fácil de localizar después de realizar la ecualización del histograma de la U-matrix. El histograma de datos muestra prácticamente la misma información que la U-matrix, aunque resuelve el problema de las esquinas de la diagonal principal. La superposición vuelve a proporcionar una imagen robusta donde es posible observar claramente los bordes amarillos entre clases, las zonas rojas asociadas a neuronas interpolantes y el grado de dispersión de las diferentes clases mediante los tonos de verde. Incluso es posible intuir que las cuatro clases representan a un número similar de patrones de entrada, ya que a medida que aumenta el área asociada a cada clase también lo hace su nivel de verde y rojo.

### 6.2.3.7. Datos reales de habla

La Figura 6.49 muestra la U-matrix (a) en tonos de verde, el histograma de datos (b) en tonos de rojo, y la superposición resultante del método (c), para un MAO ( $16 \times 16$ )-dimensional adaptado a la distribución formada por datos reales de habla de la palabra “right” (d) (proyección según las dos primeras componentes principales de la distribución). La U-matrix muestra principalmente la existencia de dos *clusters* (regiones oscuras) grandes y densos en la parte derecha de la rejilla. En la zona superior también es posible apreciar varias regiones coloreadas con un verde ligeramente oscuro, indicando la posibilidad de existencia de otras clases, algo que también ocurre en la esquina inferior-derecha. Por otro lado, la

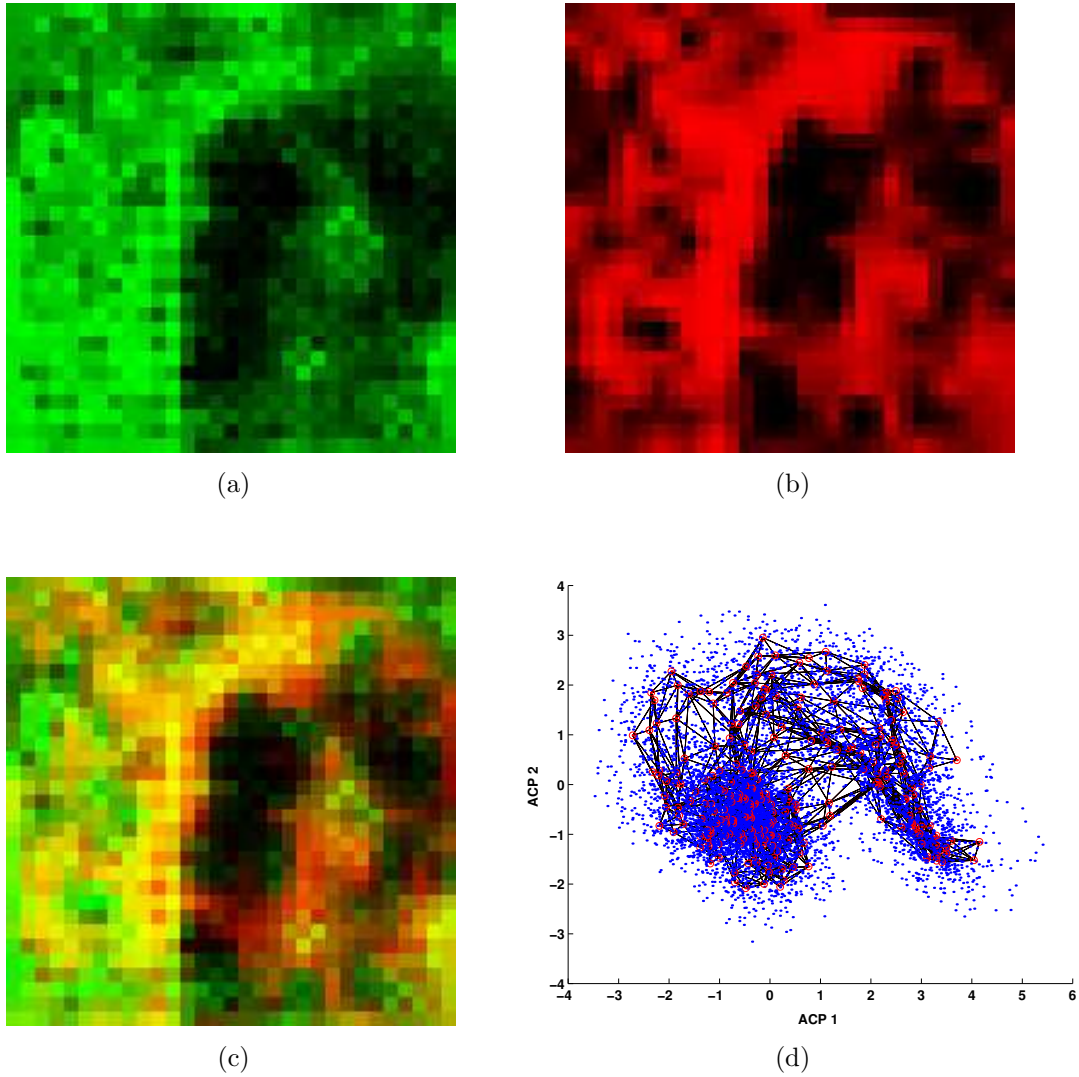


Figura 6.49: Superposición de imágenes para un MAO ( $16 \times 16$ )-dimensional adaptado a la distribución formada por datos reales de habla de la palabra “right”. U-matrix (a), histograma de datos (b), superposición (c) y proyección de los vectores de referencia conectados del MAO y de los datos de la distribución según sus dos primeras componentes principales (d).

parte izquierda de la imagen toma un color verde intenso, lo cual indica la presencia de un *cluster* disperso, datos atípicos o incluso una malformación del MAO. Posteriormente se ha comprobado que los dos *clusters* oscuros corresponden a dos tipos de silencio. De hecho, las trazas de voz utilizadas en el entrenamiento del MAO contienen al menos 50 % de silencio, por lo que era previsible encontrar una extensa región del MAO adaptada a este silencio. En realidad, este es un ejemplo de la capacidad de los MAO para realizar AED, ya que se desconocía la existencia de dos tipos diferentes de silencio antes de analizar la distribución de datos. Por otro lado, la clase dispersa situada en la parte izquierda corresponde a sonidos vocálicos. El fonema /t/ está ubicado en la esquina inferior-derecha y el /r/ en la superior-izquierda. Finalmente, la esquina superior-derecha contiene clases totalmente inesperadas antes de realizar los experimentos de análisis. Los patrones asociados a esta región proceden del final del fonema /i/, y prácticamente constituyen otro tipo de silencio. Por otro lado, en esta esquina también se localiza una clase atípica de datos procedente de una traza con un nuevo tipo de silencio. Con respecto al histograma de datos, éste se asemeja bastante a la U-matrix, aportando información similar. Las diferencias entre las dos aparecen en la superposición, donde es posible apreciar con más claridad el borde amarillo entre las vocales y el silencio; el hecho de que a pesar de que los vectores de referencia asociados al *cluster* vocálico se hallan lejos entre sí, éstos representan una cantidad elevada de patrones de entrada (por lo que la clase no es atípica ni existe una malformación del MAO); y las zonas rojizas de los *clusters* de silencio que apenas representan datos. En definitiva, la superposición genera una imagen más robusta que la U-matrix, ayudando al analista en su tarea de exploración de los datos. Esta distribución y MAO son analizados también en las Secciones 6.3 y 6.4.

### 6.2.4. Discusión

El método de superposición de imágenes surge fundamentalmente por la necesidad de mejorar la U-matrix (el método más popular de análisis visual de MAO), incorporando información sobre la densidad de datos asociada a los vectores de referencia (en este caso, mediante histogramas de datos).

El método propuesto no añade información nueva, ya que las imágenes a combinar pueden ser examinadas individualmente. No obstante, resulta ser una solución sencilla que ayuda a la interpretación de la información contenida en

los MAO. Su ventaja principal reside en que resulta mucho más fácil comparar imágenes similares cuando éstas son superpuestas una encima de otra, que compararlas ubicándolas una al lado de otra, sobre todo cuando el tamaño de éstas es elevado. Como curiosidad, el clásico pasatiempo que consiste en encontrar una serie de diferencias entre dos imágenes podría ser resuelto fácilmente aplicando el método de superposición. La posibilidad de combinar y comparar métodos que proporcionan información similar genera imágenes más completas y robustas ya que éstas aportan más información. Sin embargo, las imágenes superpuestas no deben ser analizadas individualmente. Resulta más sencillo su análisis si además las imágenes originales son visualizadas simultáneamente.

En general, la interpretación de dos colores no es complicada. Un tercer color puede incorporar demasiada complejidad, aumentando el rango de posibles colores a discriminar, que puede ser intratable. En cualquier caso, un tercer color puede ser adecuado en tareas de exploración de datos, donde el analista puede buscar colores específicos o descubrir zonas donde no esperaba encontrar ciertos colores. Los experimentos están basados en la mezcla de rojo y verde, dos colores opuestos de la teoría de colores oponentes de Ewald Hering (véase [Fai98]), aunque otras combinaciones de colores y otros modelos de color (por ejemplo, con separación uniforme entre colores) pueden favorecer el análisis de las imágenes.

Por último, el método realmente muestra vectores bidimensionales en una rejilla regular (una componente verde y otra roja). El uso del color es una solución, aunque podría ser igual de efectiva otra representación de los dos valores, por ejemplo, una representación espacial (diagramas de barras, iconos de sectores, etc.).

### 6.3. Algoritmo de Agrupación de Neuronas

Este apartado describe el tercer método propuesto en esta tesis para el análisis visual de MAO. Se trata de un algoritmo de contracción de neuronas denominado algoritmo de agrupación de neuronas (AAN) [RGG01, RG03]. Su estructura está estrechamente relacionada con la del algoritmo de entrenamiento secuencial del MAO, donde los conceptos han sido “invertidos” para poder modificar las coordenadas de las neuronas en el espacio de observación del MAO, creando una proyección no-lineal de los vectores de referencia siguiendo la filosofía de los métodos de contracción. El algoritmo modifica las posiciones de las neuronas

atrayendo aquellas cuyos vectores de referencia se hallan próximos en el espacio de los datos, sin modificar los componentes de los vectores de referencia. La fuerza de atracción está gobernada por una “tasa de aproximación” y un “núcleo de vecindad”, análogos a la tasa de aprendizaje y núcleo de vecindad del algoritmo de entrenamiento del MAO. De esta manera, es posible hablar de “inversión de conceptos”, ya que el algoritmo de entrenamiento del MAO realiza, *grosso modo*, justamente lo contrario: modifica los vectores de referencia en el espacio de los datos atrayendo aquellos cuyas neuronas asociadas se encuentran próximas en la rejilla regular (sin modificar sus posiciones), en función de una tasa de aprendizaje y un núcleo de vecindad decrecientes con respecto al tiempo.

La proyección no-lineal generada por éste algoritmo puede ser encadenada con la del MAO para generar imágenes coloreadas donde cada *cluster* aparece con un color diferente y las diferencias entre los colores dan una idea de la distancia entre los *clusters*. Esto representa una mejora notable con respecto a los métodos tradicionales de visualización de *clusters*, ya que además de ilustrar la estructura de *clusters*, las imágenes coloreadas facilitan el etiquetado y pueden ayudar a detectar discontinuidades en el MAO, por lo que también sirven como una medida visual de bondad. Por otro lado, el proceso de contracción también genera un árbol jerárquico o dendograma de clasificación “suave”, donde es posible detectar *clusters* automáticamente y ver el proceso dinámico de contracción de las neuronas.

Con respecto al método de contracción de Himberg [Him00] y otros modelos gravitacionales (véase la Sección 5.5.2.3), el algoritmo introduce una heurística diferente, que no hace uso en cada iteración de todas las neuronas a la hora de calcular sus nuevas posiciones, y está compuesto de un módulo aleatorio. De esta manera, el algoritmo gana en velocidad al presentar una complejidad computacional más ligera y puede producir diferentes proyecciones no-lineales (similares), lo cual puede ser útil desde un punto de vista exploratorio. En los experimentos llevados a cabo las proyecciones de las distribuciones son similares a las obtenidas con el método de Himberg, aunque el aumento en la velocidad del algoritmo y su naturaleza aleatoria hace disminuir ligeramente su precisión a la hora de mostrar dichas distribuciones. En cualquier caso, esta pérdida de precisión no ha resultado significativa en los experimentos llevados a cabo.

### 6.3.1. Descripción del problema

Los métodos tradicionales de visualización de MAO son capaces de revelar información importante de las distribuciones de datos utilizadas en su entrenamiento. Éstos aprovechan la estructura regular del MAO para describir información exacta y objetiva, como distancias entre vectores de referencia, proporción de datos asociados a cada neurona, valor de los componentes de los vectores de referencia, etc. Por el contrario, los algoritmos de contracción y coloración son más sofisticados y utilizan heurísticas menos precisas y rigurosas para representar la distribución de los vectores de referencia adaptados a los datos y su estructura de *clusters*. Sin embargo, a pesar de la falta de rigor de estos métodos, son capaces de proporcionar imágenes con las que el análisis de MAO es mejorado notablemente, al aportar información más rica sobre los datos. En concreto, los métodos de contracción generan:

- Proyecciones no-lineales de los vectores de referencia.
  - Proporcionan una visión general de los vectores de referencia no asociada a la estructura regular del MAO.
  - Proporcionan un dendograma o árbol jerárquico de clasificación suave, el cual puede ser cortado por varios niveles para generar diferentes proyecciones.
  - Permiten la detección (manual y/o automática) de *clusters*.
  - Permiten identificar clases atípicas y neuronas interpolantes.
- Imágenes coloreadas de la estructura de *clusters*.
  - Permiten la detección de los diferentes *clusters* y los bordes entre ellos.
  - Facilitan el análisis de trayectorias.
  - Facilitan el etiquetado de las neuronas.
  - Aportan una medida visual de bondad del MAO.

En los experimentos, el método de contracción de Himberg es capaz de generar imágenes que representen las distribuciones de datos de forma precisa. Sin embargo, su complejidad computacional es elevada (véase la Sección 5.5.2.3), ya que utiliza todas las neuronas en cada iteración para calcular las sucesivas posiciones



de las neuronas. El objetivo del nuevo método consiste en la creación de un algoritmo de contracción eficiente, de menor complejidad computacional, capaz de generar imágenes similares a las producidas por el método de Himberg.

Una primera idea consiste en crear una variante del propio algoritmo de Himberg de modo que sólo actualice un conjunto limitado de neuronas en cada iteración. De esta manera, sería posible reducir el número de operaciones del algoritmo ignorando la fuerza de atracción de neuronas cuyos vectores de referencia se hallan lejos entre sí, y que por tanto no influyen significativamente a la hora de calcular los desplazamientos de las neuronas. Por ejemplo, los pesos  $s_{i,j}$  de la suma ponderada (de la matriz  $S$ ) podrían ser anulados si la distancia  $\|w_i - w_j\|$  resulta ser lo suficientemente grande como para no afectar al cómputo total de los desplazamientos de la neuronas. La cantidad de elementos de  $S$  anulados debe permanecer fija, con lo cual la proyección converge a un solo punto ( $S^r X_0, r \rightarrow \infty$ ), véase la Sección 5.5.2.3. En caso de aumentar progresivamente según aumenta el número de iteraciones del algoritmo, la matriz  $S$  sería modificada en cada iteración de (5.15), quedando  $X_{i+1} = S_i X_i, i = 0, 1, \dots, r$ . En este último caso la proyección no converge a un solo punto y el algoritmo es más lento al tener que calcular una nueva matriz  $S_i$  en cada iteración. Con estas variantes se pretende reducir la complejidad computacional del método. Si en el original ésta se aproxima a  $\mathcal{O}(RM^2)$  ( $R$  = número de iteraciones del algoritmo,  $M$  = número de neuronas del MAO), con estas variantes podría ser reducida a  $\mathcal{O}(KRM)$ , donde  $K$  depende de  $M$  y puede llegar a ser mucho menor ( $K \ll M$ ). En realidad, estas ideas surgieron como consecuencia de los trabajos relacionados con el AAN propuesto en este apartado, con los que es posible obtener buenos resultados (similares a los del método de Himberg original) utilizando conjuntos reducidos de neuronas a la hora de calcular las sucesivas posiciones de éstas. Sin embargo, estas variantes no han sido probadas hasta ahora, por lo que se propone su implementación, estudio y comparación con otros algoritmos de contracción como una línea futura de esta tesis (véase el Capítulo 7).

Sin embargo, no sólo interesa incrementar la velocidad del algoritmo, sino que también se busca un método basado en una heurística de atracción diferente, que introduzca una nueva filosofía de organización, y que presente varias propiedades generales diferentes de las del método de Himberg y de los métodos gravitacionales como el de Wright.

Con respecto a las cuatro propiedades del algoritmo gravitacional de Wright (véase la Sección 5.5.2.3), la única que debe ser respetada en el nuevo algoritmo es la de mantener un parámetro que limite el movimiento de las neuronas en el espacio de salida. Esto es necesario para evitar que las neuronas se aproximen a demasiada velocidad. La fuerza de atracción debe ser moderada para evitar que las neuronas se crucen o “pasen de largo”, para asegurar que el método no diverja (que las neuronas no se alejen) y para evitar otros efectos como los oscilatorios. Por otro lado, la fuerza de atracción debe permitir analizar el proceso de contracción detalladamente.

Las otras tres propiedades no son estrictamente necesarias. Al querer reducir la complejidad computacional, es necesario trabajar con conjuntos reducidos de neuronas. Es decir, no todas las neuronas deben intervenir a la hora de calcular sus nuevas posiciones. Esto puede afectar ligeramente a la precisión del método. Sin embargo, teniendo en cuenta que el método va a estar basado en una heurística y que las proyecciones normalmente no preservan todas las distancias originales, las imágenes siempre van a contener defectos al proyectar datos de dimensión elevada sobre un espacio de menor dimensión. Desde este punto de vista, en la práctica el grado de precisión no resulta ser un factor importante. Por ejemplo, si en los datos existe un *cluster* de neuronas, el objetivo principal del método debe ser mostrarlo con claridad, aunque su posición relativa con respecto a otros *clusters* o su forma proyectada se alejen ligeramente de unas supuestas óptimas.

Tampoco es necesario que la proyección converja a un solo punto, ya que naturalmente dicha proyección no aporta información válida. Lo interesante del método debe consistir en analizar el proceso dinámico de atracción u organización de las neuronas y varias proyecciones intermedias estáticas que revelen estructuras en los datos (*clusters*, clases atípicas, relaciones en los datos, etc.). Por último, al igual que en el método de Himberg, no es posible implementar una fusión de dos neuronas ya que los cálculos están basados en las distancias entre sus vectores de referencia, que generalmente van a ser distintos.

## 6.3.2. Solución propuesta

### 6.3.2.1. Inversión de conceptos

El algoritmo de entrenamiento secuencial del MAO resulta sencillo de implementar y simular, y sus propiedades prácticas son claras y fáciles de observar. Sin

embargo, sus propiedades teóricas siguen sin ser demostradas en el caso general, a pesar del enorme esfuerzo de varios autores [CFP98]. Quizás por esta razón el autor de esta tesis no ha encontrado trabajos que aprovechen su estructura algorítmica para resolver otros problemas de auto-organización diferentes al del propio entrenamiento del MAO.

La clave para entender el algoritmo propuesto en este apartado reside en entender el proceso de auto-organización del MAO desde un punto de vista general. El algoritmo de entrenamiento del MAO puede ser visto como un proceso en el que una serie de elementos son organizados (vectores de referencia, los cuales además se adaptan a unos datos de entrada) en un espacio “primario”, en función de unas disimilitudes entre ellos definidas en otro espacio “secundario” (las distancias entre sus respectivas neuronas en la rejilla regular del MAO). Es decir, un conjunto de elementos del espacio primario son organizados siguiendo, aproximadamente, la distribución u organización de otro conjunto de elementos asociados en un espacio secundario (existiendo una correspondencia uno-a-uno entre ambos conjuntos). La tasa de aprendizaje y el núcleo de vecindad son escogidos adecuadamente para que el algoritmo “converja” y para conseguir una organización de esos elementos en el espacio primario similar a la definida en el secundario. Al principio los elementos tienen mucha movilidad para conseguir la organización “global” deseada y progresivamente se limita su movimiento para suavizar el proceso y favorecer un ordenamiento “local” (y para que se ajusten a los datos de entrada, en el caso de los vectores de referencia de un MAO).

Ahora bien, no es necesario que los elementos del espacio secundario estén organizados en una estructura regular (ni incluso que éste sea un espacio métrico). Por otro lado, si deseamos que la organización de los elementos del espacio primario se aproxime a la del secundario, es necesario prescindir del ajuste de dichos elementos a unos datos de entrada.

Considérese el siguiente problema (no-trivial) de auto-organización: sea un recinto (espacio primario) en el que se encuentran varias personas de diferentes nacionalidades, las cuales se hallan ubicadas inicialmente al azar. Se desea organizarlas dentro de tal recinto en función de sus lugares de nacimiento (localización geográfica en un espacio secundario). Una solución a este problema puede consistir en aplicar una estrategia basada en un modelo de contracción. Cada persona ejercería una fuerza de atracción hacia sí sobre el resto de personas inversamente

proporcional a la distancia entre sus lugares de nacimiento (por ejemplo, la longitud de una geodésica en la “esfera” asociada al globo terráqueo). Al final del proceso aparecería en el recinto un mapa mundial distorsionado, pero organizado (especialmente si antes de comenzar el proceso existiese una cierta organización geográfica de las personas en el recinto).

Este mismo problema puede ser resuelto utilizando un algoritmo con una estructura similar a la del entrenamiento secuencial del MAO. Considérese un algoritmo iterativo donde en cada iteración una persona del recinto es seleccionada al azar para atraer al resto de personas hacia ella. La fuerza de atracción que ejerce dicha persona sobre otra podría estar determinada por una tasa de “atracción” o “aproximación” (decreciente con el tiempo), por la distancia que los separa en el recinto y por un factor (también decreciente con el tiempo) inversamente proporcional a la distancia entre sus lugares de nacimiento (por ejemplo, un núcleo de vecindad centrado en el lugar de nacimiento de la persona elegida donde toma su valor máximo, que decrece a medida que aumenta la distancia con respecto a ese centro). De esta manera, las personas nacidas en una zona geográfica cercana aparecerán cerca en el recinto al final del proceso, y al igual que en el caso anterior, en el recinto aparecerá un mapa mundial distorsionado, pero organizado. Por otro lado, nótese que la similitud entre las personas (espacio secundario) no tiene que estar necesariamente basada en características espaciales. Ésta puede estar definida en función de diversos rasgos, que pueden ser representados mediante todo tipo de variables (continuas, discretas, difusas, binarias, nominales, ordinales, etc.).

Volviendo a los MAO, las proyecciones no-lineales generadas por los métodos de contracción pueden ser vistas como reorganizaciones de las neuronas en el espacio de observación del MAO según el esquema descrito anteriormente. Nótese que el objetivo consiste en organizar un conjunto de elementos (las neuronas) en un espacio primario (el espacio de observación del MAO) según la distribución de un conjunto de elementos asociados (los vectores de referencia), en un espacio secundario (el espacio de los datos). Es decir, es posible aplicar un algoritmo similar al del entrenamiento del MAO en sentido contrario, donde los conceptos son invertidos, para conseguir la deseada proyección no-lineal. Partiendo de la configuración regular inicial de las neuronas, ésta puede ser modificada para conseguir una proyección lineal que intente reflejar la estructura de los datos, donde las distancias entre los vectores de referencia proyectados (neuronas) sean

preservadas mejor que en la rejilla regular.

La Figura 6.50 ilustra la idea de la inversión de conceptos, donde los papeles de las neuronas y los vectores de referencia son “intercambiados” para obtener una reorganización de las neuronas en función de la configuración de los vectores de referencia. En el entrenamiento del MAO (en este caso  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro) los vectores de referencia se adaptan a los datos en un espacio primario (a), pero están organizados en función de las posiciones fijas de sus respectivas neuronas en la rejilla regular de un espacio secundario (b). En el proceso contrario, son las neuronas las que se mueven en un nuevo espacio primario (c), para quedar organizadas de acuerdo con la configuración de los vectores de referencia calculados en el entrenamiento del MAO (d), que ahora permanecen fijos en un espacio secundario. La imagen en (c) ha sido calculada mediante el AAN.

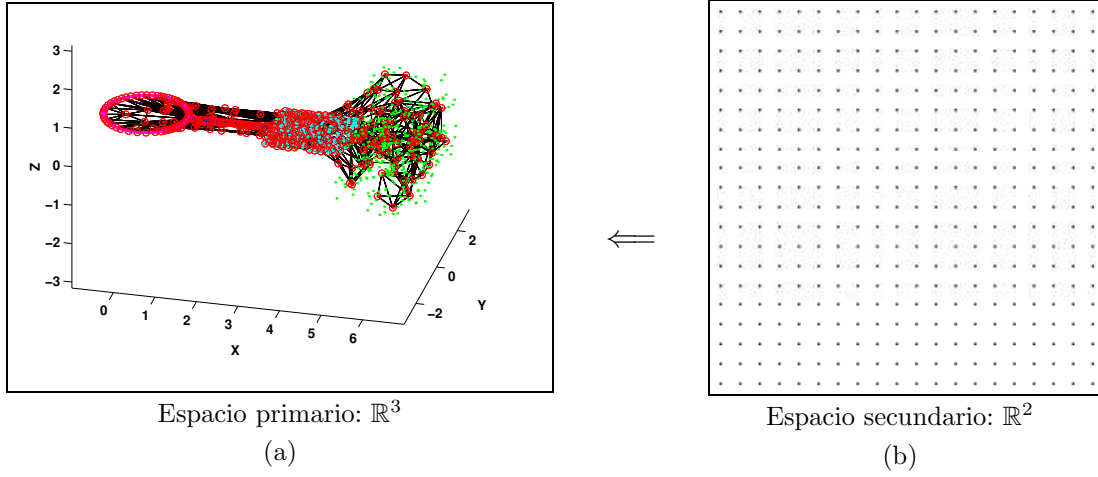
#### 6.3.2.2. Algoritmo

Formalmente, el AAN queda definido mediante los siguientes pasos:

1. Escoger una neurona elegida.
2. Actualizar un conjunto de neuronas "vecinas" de la elegida en dirección a ésta, según una tasa de aproximación y un núcleo de vecindad centrado en el vector de referencia de la neurona elegida (y opcionalmente también en la propia neurona).
3. Si se ha procesado un número determinado de muestras  $T$ : fin; en caso contrario, reducir la tasa de aproximación y el radio de vecindad, e ir al paso 1.

Véase la similitud del algoritmo con el del entrenamiento secuencial del MAO en la Sección 3.2.4.1, donde la mayor diferencia reside en eliminar la actualización de las neuronas hacia una serie de datos. En realidad, el algoritmo podría ser modificado ligeramente para que la neurona “elegida” fuera calculada y no seleccionada al azar, en función de las distancias entre éstas y un conjunto de datos de

### Entrenamiento del MAO



### Proceso contrario

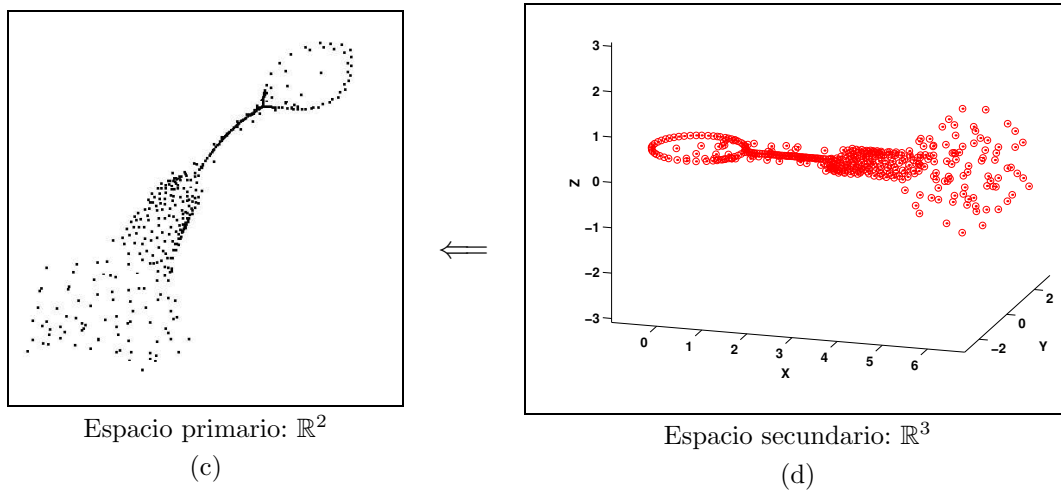


Figura 6.50: Inversión de conceptos. En el entrenamiento del MAO los vectores de referencia son ajustados a los datos en un espacio primario (a), en función de la configuración regular de éstos en un espacio secundario (b). El proceso contrario consiste en actualizar las posiciones de las neuronas (imagen obtenida a partir del AAN), ahora en un espacio primario (c), en función de la configuración de los vectores de referencia en un espacio secundario (d). El MAO ( $20 \times 20$ )-dimensional utilizado ha sido adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro.

entrada proyectados sobre el propio espacio de observación o mapa. En este caso, el algoritmo sería una versión “invertida” exacta del algoritmo de entrenamiento del MAO. Las neuronas serían adaptadas a los datos proyectados, pero siguiendo la organización definida por los vectores de referencia en el espacio de los datos. Sin embargo, esto daría lugar a proyecciones de las neuronas cuya utilidad no es clara. Afortunadamente, el objetivo del algoritmo propuesto es crear un algoritmo de contracción, por lo que es posible prescindir de las proyecciones de los datos de entrada.

El primer paso consiste en seleccionar una neurona “elegida”  $\eta_c$ , cuya posición en el espacio de observación es  $r_c$  y cuyo vector de referencia asociado es  $w_c$ . En los experimentos los resultados escogiéndola al azar han resultado ser aceptables, por lo que no parece necesario utilizar una estrategia más compleja. En cualquier caso, no se descarta el uso y estudio de otras opciones a la hora seleccionar esta neurona, como ciclos aleatorios u otras estrategias no-aleatorias.

El segundo paso consiste en atraer hacia la neurona elegida a un conjunto de neuronas “vecinas” de ésta según la siguiente regla:

$$r_i(t+1) = r_i(t) + \lambda(t)\psi_{c,i}(t)[r_c(t) - r_i(t)] \quad (6.15)$$

donde  $\lambda(t)$  es la “tasa de aproximación” y  $\psi_{c,i}(t)$  un núcleo de vecindad, los cuales pueden ser análogos a la tasa de aprendizaje  $\alpha(t)$  y al núcleo de vecindad  $h_{c,i}(t)$ , respectivamente, de (3.3). El producto de ambas determina la proporción de la distancia entre  $r_c(t)$  y  $r_i(t)$  que va a recorrer la neurona  $\eta_i$  en dirección a la  $\eta_c$ , por lo que debe ser menor o igual a uno y decrecer hasta cero según se suceden las iteraciones  $(t)$  del algoritmo.

La tasa de aproximación  $\lambda(t)$  es una función monótona decreciente que tiende a cero, y cuya elección no es crítica (puede ser lineal, exponencial o inversamente proporcional a  $t$ ). Los experimentos utilizan las dos siguientes tasas de aproximación:

$$\lambda(t) = \lambda(0) \cdot \left(1 - \frac{t}{T}\right) \quad (6.16)$$

y

$$\lambda(t) = \lambda(0) \cdot \exp\left(-\frac{2t}{T}\right) \quad (6.17)$$

donde el parámetro  $T$  representa el número total de iteraciones que efectúa el

algoritmo y  $\lambda(0)$  es el valor inicial de la tasa de aproximación. La tasa de aproximación es prácticamente idéntica que la tasa de aprendizaje del algoritmo de entrenamiento del MAO.

En cuanto al núcleo de vecindad  $\psi_{c,i}(t)$  existe una diferencia importante con respecto al algoritmo de entrenamiento del MAO: éste puede ser definido en el espacio de los datos y también en el de observación. Es decir, debe definir un conjunto de neuronas cuyos vectores de referencia se hallan cerca del de la elegida en el espacio de los datos, pero también puede restringir ese conjunto para que las neuronas también se encuentren alrededor de la elegida en el mapa. Por tanto, los experimentos utilizan dos núcleos de vecindad diferentes, dependiendo de la imposición de esta última restricción. Posteriormente, cada neurona perteneciente a ese conjunto será atraída hacia la elegida en función de la distancia entre sus vectores de referencia  $\|w_c - w_i\|$ , y de otras variables relacionadas con un “radio de vecindad”, que serán descritas a continuación. Naturalmente, junto con la tasa de aproximación,  $\psi_{c,i}(t)$  contribuye a establecer la fuerza de atracción de la neurona elegida, la cual debe tender a cero a medida que aumenta  $\|w_c - w_i\|$ .

El primer núcleo de vecindad empleado está estrechamente relacionado con el descrito en (3.7):

$$\psi_{c,i}(t) = \exp\left(-\frac{\|w_c - w_i\|^2}{2\sigma^2(t)}\right) \quad (6.18)$$

La única diferencia reside en que está centrado en el vector de referencia (en el espacio de los datos) de la neurona elegida  $\eta_c$  y depende de distancias entre vectores de referencia. Naturalmente, la fuerza de atracción es mayor cuanto menor es la distancia  $\|w_c - w_i\|$ . Asimismo, depende implícitamente de un “radio de vecindad”  $\sigma(t)$ , que es otra función monótona decreciente que puede ser definida de forma análoga a la tasa de aproximación o tasa de aprendizaje del algoritmo de entrenamiento del MAO. La elección del radio de vecindad inicial  $\sigma(0)$  es importante en el proceso ya que al principio el núcleo de vecindad debe contener un porcentaje relativamente elevado de vectores de referencia. Los experimentos utilizan generalmente la versión lineal decreciente de  $\sigma(t)$ , donde  $\sigma(0)$  corresponde al 40 % del diámetro del conjunto de vectores de referencia (máxima distancia entre todos los pares de vectores de referencia):

$$\sigma(t) = \sigma(0) \cdot \left(1 - \frac{t}{T}\right) \quad (6.19)$$



donde  $\sigma(0) = \tau \cdot \mathcal{D}$ , siendo  $\tau \in (0, 1]$  una constante y  $\mathcal{D} = \max\{\|w_i - w_j\|\}$ ,  $i, j = 1, \dots, M$ .

El segundo núcleo de vecindad utilizado en los experimentos es definido mediante:

$$\psi_{c,i}(t) = \chi_{c,i}(t) \cdot \exp\left(-\frac{\|w_c - w_i\|}{\beta(t) \cdot \delta_c}\right) \quad (6.20)$$

La diferencia principal con respecto a (6.18) radica en la función  $\chi_{c,i}(t)$ , que es un núcleo de vecindad binario centrado en la neurona  $\eta_c$ , el cual define la restricción de que las neuronas a adaptar se hallen cerca de la elegida:

$$\chi_{c,i}(t) = \begin{cases} 0 & \text{si } \|r_c - r_i\| > \sigma(t) \\ 1 & \text{si } \|r_c - r_i\| \leq \sigma(t) \end{cases} \quad (6.21)$$

donde en este caso  $\sigma(t)$  es un radio de vecindad definido en el espacio de observación del MAO. Se trata de una función monótona decreciente que puede ser definida de forma análoga a (6.19), con la diferencia de que en este caso debe ser tomada considerando las dimensiones del espacio de observación, donde es conveniente que el radio inicial  $\sigma(0)$  sea mayor que la mitad del diámetro del MAO.

El resto de la función  $\psi_{c,i}(t)$  es una ligera variante de (6.18). La distancia  $\|w_c - w_i\|$  no aparece elevada al cuadrado,  $\beta(t)$  es una función lineal decreciente:

$$\beta(t) = \beta(0) + [\beta(T) - \beta(0)] \cdot \frac{t}{T} \quad (6.22)$$

los experimentos generalmente utilizan  $\beta(0) = 1/10$  y  $\beta(T) = 1/15$ . Finalmente,  $\delta_c$  corresponde a la distancia máxima desde el vector de referencia de la neurona elegida al resto de vectores de referencia del MAO:  $\delta_c = \max\{\|w_c - w_i\|\}$ ,  $i = 1, \dots, M$ . De esta manera, en general, una neurona cuyo vector de referencia se halle lejos de la media de vectores de referencia (posiblemente ubicada cerca del perímetro del MAO) tendrá una fuerza de atracción mayor que otra cuyo vector de referencia esté localizado más cerca de esta media. Esto es implementado para impedir un colapso de las neuronas hacia el centro del espacio de observación (nótese que también puede ser definida en el espacio de observación  $\delta'_c = \max\{\|r_c - r_i\|\}$ ,  $i = 1, \dots, M$ ).

El algoritmo puede ser clasificado como método de contracción ya que el proceso básicamente atrae a varias neuronas hacia otra que permanece fija en

cada iteración. Por tanto, el método comparte las características generales de este tipo de métodos. Además de la proyección no-lineal, que aporta información por sí sola, ésta puede ser encadenada con un código de colores para obtener una imagen coloreada de la estructura de *clusters* de la distribución, como describe la Sección 5.5.2.

### 6.3.2.3. Complejidad computacional

La complejidad computacional del AAN es especialmente importante ya que en ella radica la ventaja principal sobre el método de contracción de Himberg, cuya complejidad computacional se acerca a  $\mathcal{O}(RM^2)$ , donde  $R$  es el número de iteraciones del algoritmo y  $M$  representa al número de neuronas del MAO. El método de Himberg también puede ser ejecutado en una sola iteración si no se desea analizar el proceso de contracción, en cuyo caso su complejidad es reducida a  $\mathcal{O}(M^2)$ , véase la Sección 5.5.2.3.

La complejidad computacional del AAN es  $\mathcal{O}(TM)$ , donde  $T$  es el número de iteraciones del algoritmo y  $M$  el número de neuronas del MAO. A continuación se analizan éstos dos factores que influyen en la velocidad del algoritmo:

- Número de iteraciones del algoritmo ( $T$ ). Es uno de los parámetros que debe escoger el analista en función de las estructuras y características de la distribución de datos a analizar. En la práctica no es necesario escoger un valor elevado para este parámetro. De hecho, es posible obtener resultados aceptables con  $T = K\sqrt{M}$ , donde  $K$  es una constante relativamente pequeña ( $5 \leq K \leq 10$ ) y  $M$  es el número de neuronas del MAO. No obstante, este parámetro influye en la precisión del algoritmo, la cual parece ser mayor si todas las neuronas influyen de igual manera en el proceso de contracción, como en el método de Himberg. Por la ley de los grandes números, a medida que aumenta el número de iteraciones  $T$  las frecuencias con las que las neuronas son “elegidas” tienden a igualarse, y por tanto a igualarse la influencia de cada neurona en el proceso, aumentando la precisión final de la proyección. Cuando el número de iteraciones  $T$  es elevado la tasa de aproximación inicial  $\lambda(0)$  debe ser escogida pequeña para evitar un colapso de las neuronas al centro del mapa. En cualquier caso, el objetivo del AAN no es obtener proyecciones muy precisas sino eficientes, con lo que debe ser aplicado con un valor de  $T$  bajo o moderado.

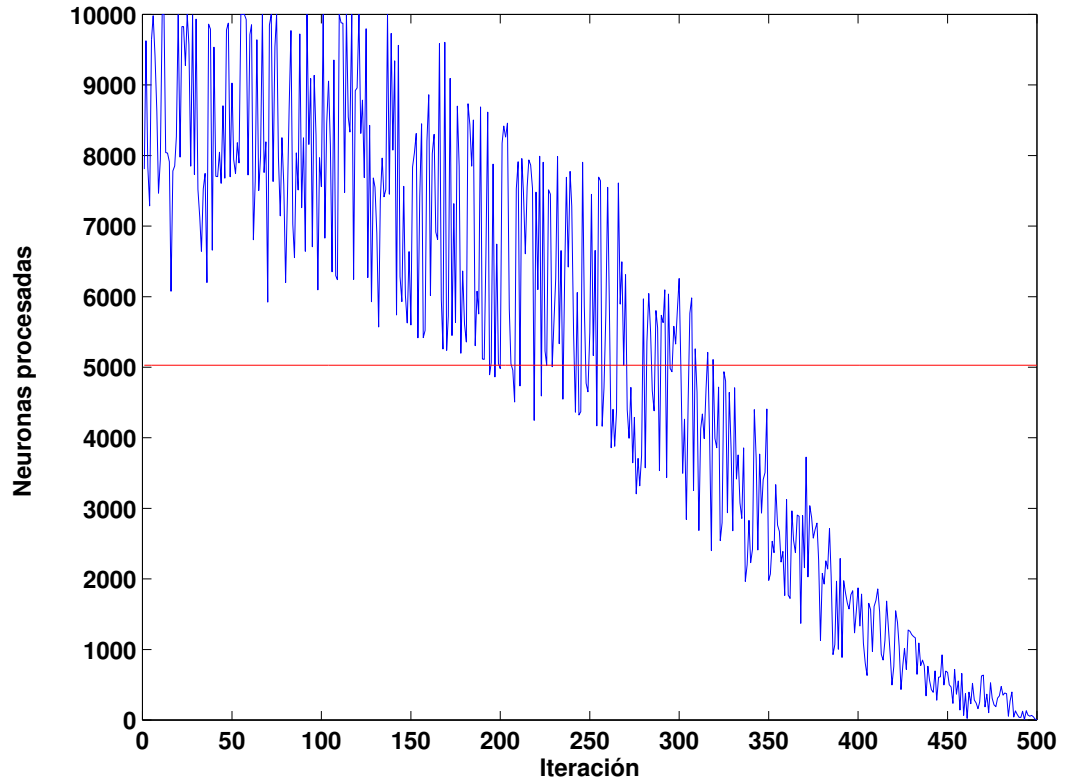


Figura 6.51: Partiendo de un MAO  $(100 \times 100)$ -dimensional (adaptado a la distribución formada por dos eslabones encadenados) la gráfica muestra el número de neuronas procesadas en las 500 iteraciones necesarias para ejecutar el AAN, el cual decrece a medida que avanza el algoritmo, de manera que la media de neuronas procesadas en cada iteración (representada mediante la línea roja horizontal) es aproximadamente 5000.

- Número de neuronas del MAO ( $M$ ). Este factor corresponde al número de neuronas que deben ser procesadas en cada iteración del algoritmo. Como al principio del proceso suelen intervenir prácticamente todas las neuronas, en la complejidad computacional debe aparecer  $M$ , que es el valor máximo de neuronas a procesar. Sin embargo, a medida que avanza el proceso, el número de neuronas a tratar y desplazar disminuye considerablemente, por lo que al final del proceso sólo se habrá procesado una fracción (aproximadamente un 50 %, véase la Figura 6.51) del producto  $TM$ . En otras palabras, la media de neuronas tratadas en cada iteración suele ser aproximadamente la mitad de las del MAO.

Teniendo en cuenta estas consideraciones, la complejidad computacional del algoritmo se aproxima a  $\mathcal{O}(K\sqrt{M} \cdot M/2)$ , es decir, aproximadamente  $\mathcal{O}(M^{3/2})$ .

La Figura 6.51 muestra una gráfica que representa el número de neuronas

procesadas en las 500 iteraciones necesarias para ejecutar el AAN para un MAO  $(100 \times 100)$ -dimensional (adaptado a la distribución formada por dos eslabones encadenados). Obsérvese que  $M = 10000$  y que el número de iteraciones  $T$  corresponde al valor  $5\sqrt{M}$ . La gráfica ilustra como el número de neuronas procesadas decrece a medida que avanza el algoritmo, lo cual lo acelera en las últimas iteraciones. El algoritmo procesa en total 2514441 neuronas ( $\approx 50\%$  del producto  $TM$ ), donde la media de neuronas procesadas en cada iteración es aproximadamente  $M/2 = 5000$ . La Sección 6.3.4 presenta un estudio sobre la velocidad del AAN y el método de contracción de Himberg.

### 6.3.3. Resultados experimentales: proyecciones

A continuación varios experimentos son descritos con el objetivo de ilustrar las características del AAN, analizar sus ventajas y limitaciones, y compararlo especialmente con el método de contracción de Himberg (véase la Sección 5.5.2.3). La Sección 6.3.4 analiza la velocidad del AAN y del método de Himberg.

Los MAO utilizados son de tamaño medio o grande (generalmente  $(20 \times 20)$ -dimensionales) ya que generalmente no es posible representar adecuadamente una distribución de datos con un conjunto reducido de vectores de referencia. Varias figuras de los apartados anteriores muestran la adaptación de éstos a los datos. La forma de los MAO es cuadrada, por lo que se ha empleado el código de colores basado en el plano  $B = 0$  (cuadrado) del cubo RGB, que parece producir descripciones más precisas que el basado en el plano  $B = 255 - G$  (rectangular).

#### 6.3.3.1. Parámetros utilizados

Afortunadamente, el conjunto de parámetros que intervienen en la regla de actualización de las neuronas (6.15) no juega un papel crítico en el proceso, a excepción del radio inicial del núcleo de vecindad, que no debe ser pequeño. De esta manera, no ha habido necesidad de adaptar todos los parámetros a las características de cada MAO y distribución a analizar, que *a priori* son desconocidos. Por esta razón, en prácticamente todos los experimentos se ha utilizado el mismo conjunto de parámetros “estándar”, dependiendo del tipo de núcleo de vecindad utilizado en (6.15). Los dos principales conjuntos de parámetros utilizados son descritos a continuación en función del núcleo de vecindad empleado:

- Núcleo de vecindad descrito en (6.18):

- Tasa de aproximación  $\lambda(t)$  lineal (según (6.16)), donde su valor en la primera iteración del algoritmo es  $\lambda(0) = 0,2$ .
  - Radio de vecindad  $\sigma(t)$  lineal (según (6.19)), donde su valor en la primera iteración del algoritmo es el 40 % del diámetro del conjunto de vectores de referencia. Es decir,  $\sigma(0) = \tau \cdot \mathcal{D}$ , donde  $\tau = 0,4$  y  $\mathcal{D} = \max\{\|w_i - w_j\|\}, i, j = 1, \dots, M$ .
  - Número de iteraciones  $T = 200$  para MAO  $(20 \times 20)$ -dimensionales o de menor tamaño.
- Núcleo de vecindad descrito en (6.20):
    - Tasa de aproximación  $\lambda(t)$  lineal (según (6.16)), donde su valor en la primera iteración del algoritmo es  $\lambda(0) = 0,8$ .
    - Radio de vecindad  $\sigma(t)$  lineal (según (6.19)), donde su valor en la primera iteración del algoritmo  $\sigma(0)$  es aproximadamente la longitud de una diagonal del mapa en el espacio de observación (para abarcar todas las neuronas al principio del proceso). En los experimentos las imágenes generadas son cuadrados de  $(500 \times 500)$  pixeles, por lo que  $\sigma(0) = 700 \approx 500\sqrt{2}$ .
    - Función  $\beta(t)$  lineal (según 6.22), donde  $\beta(0) = 1/10$  y  $\beta(T) = 1/15$ .
    - Número de iteraciones  $T = 200$  para MAO  $(20 \times 20)$ -dimensionales o de menor tamaño, y  $T=500$  para MAO  $(100 \times 100)$ -dimensionales.

### 6.3.3.2. Cuatro *clusters* disjuntos con forma hipercúbica

Los primeros ejemplos del ANN utilizan un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos con forma hipercúbica, véase la Figura 6.52. Las Figuras 6.53 y 6.54 muestran dos tipos de proyecciones que genera el algoritmo y el correspondiente encadenado basado en un código de colores lineal con la proyección original regular del MAO utilizando los núcleos de vecindad definidos en (6.18) y (6.20), respectivamente. Mediante estos ejemplos se persigue analizar el comportamiento general del AAN con un caso sencillo, y comparar los resultados en función del tipo de núcleo de vecindad utilizado. Los parámetros utilizados corresponden a los descritos en la Sección 6.3.3.1, a excepción del número de iteraciones totales  $T$  que en este caso sólo es 100. La

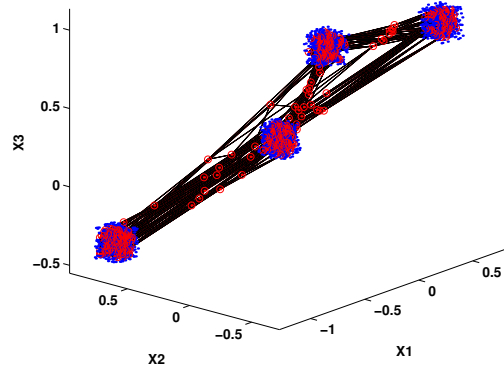


Figura 6.52: MAO  $(20 \times 20)$ -dimensional adaptado a la distribución 8-dimensional formada por cuatro *clusters* disjuntos con forma hipercúbica (sólo los tres primeros componentes son visualizados).

reducción del número de iteraciones es posible dada la relativa simplicidad de la distribución de datos, la cual contiene *clusters* claramente separados (disjuntos) del mismo tamaño. En ambos ejemplos la proyección original del MAO ( $t = 0$ ) y el código de colores lineal (plano  $B = 0$  del cubo RGB) corresponden a los ilustrados en la Figura 5.39 (a). Las imágenes de las Figuras 6.53 y 6.54 han sido obtenidas en los instantes o iteraciones  $t = 15, 40$  y  $100$ , en (a), (c) y (e), respectivamente. Nótese la similitud entre las proyecciones obtenidas y las ilustradas en la Figura 5.39, la cual muestra el mismo ejemplo basado en las proyecciones del método de Himberg. El AAN consigue atraer a las neuronas que componen cada *cluster* para formar progresivamente grupos compactos de puntos, mientras las neuronas interpolantes aparecen claramente como puntos aislados entre los *clusters*. Por otro lado, las imágenes coloreadas también son muy similares a las obtenidas en la Figura 5.39 (y a las generadas mediante métodos tradicionales de visualización, véase la Figura 6.55). En ambos ejemplos los encadenados basados en un código de colores lineal de las proyecciones de (a), (c) y (e) con la proyección original del MAO son ilustrados en (b), (d) y (f), respectivamente. En estas imágenes es posible detectar los cuatro *clusters* con claridad a través de su color asignado. Éste aparece más uniforme a medida que aumenta  $t$ , y las neuronas interpolantes también aparecen con mayor claridad. Comparando las dos proyecciones obtenidas, la generada con el núcleo de vecindad descrito en (6.18) tiene más facilidad para mantener la estructura regular de las posiciones de las

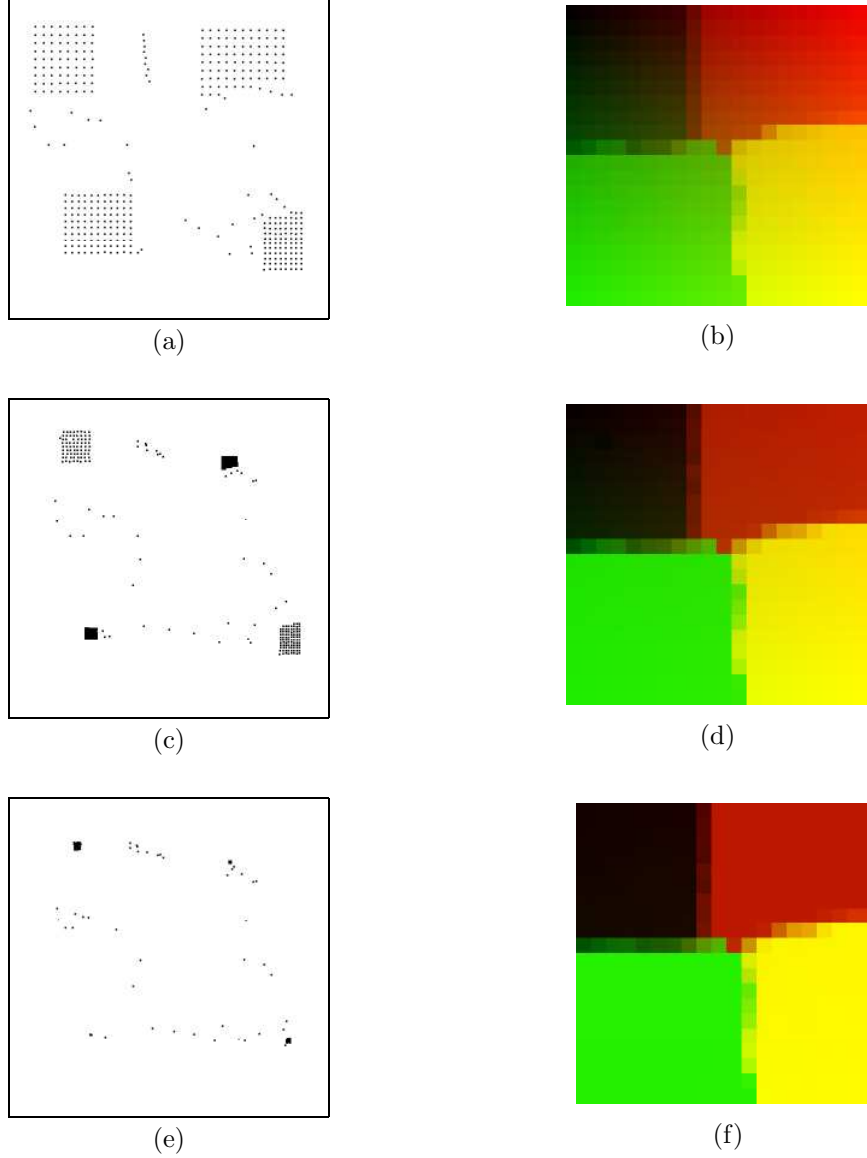


Figura 6.53: Proceso de contracción del AAN basado en el núcleo de vecindad descrito en (6.18) y encadenados basados en un código de colores lineal para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos con forma hipercúbica. Las proyecciones son generadas en los instantes  $t = 15, 40$  y  $100$ , en (a), (c) y (e), respectivamente, y los encadenados en (b), (d) y (f), respectivamente.

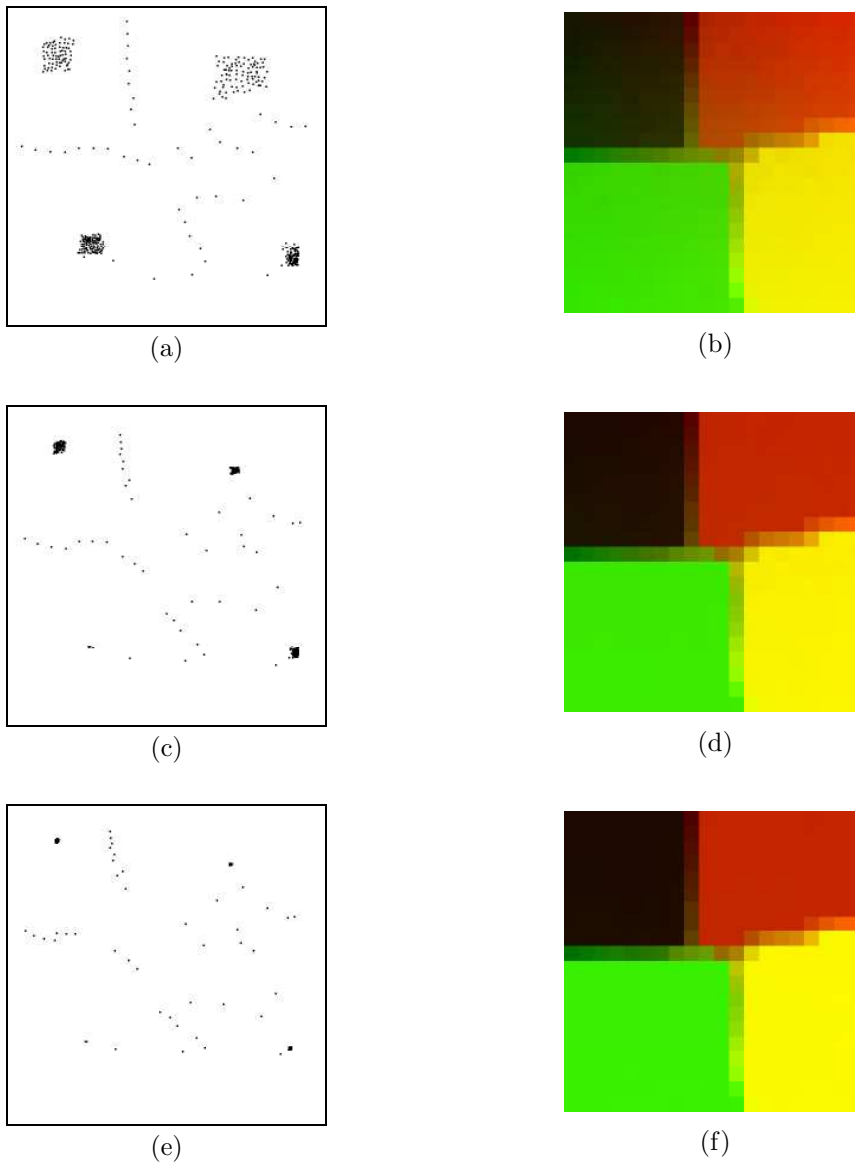


Figura 6.54: Mismo ejemplo que el presentado en la Figura 6.53 utilizando el núcleo de vecindad descrito en (6.20).



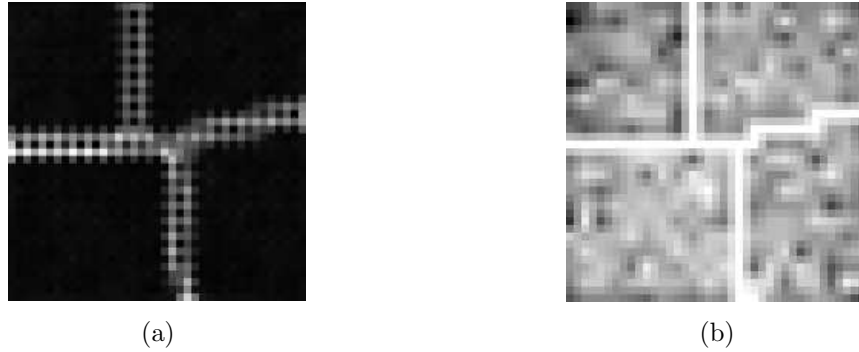


Figura 6.55: U-matrix (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos.

neuronas. Sin embargo, esta característica generalmente no es deseable, ya que los datos no suelen aparecer organizados en estructuras regulares. Por esta razón, y a pesar de que las imágenes finales suelen producir proyecciones similares, el método generalmente representa mejor las distribuciones y los procesos dinámicos de contracción utilizando el núcleo de vecindad descrito en (6.20).

### 6.3.3.3. Ocho *clusters* disjuntos con forma hipercúbica

El siguiente experimento muestra un ejemplo de la proyección del AAN y la imagen coloreada de la estructura de *clusters* para la distribución formada por ocho *clusters* disjuntos con forma hipercúbica, a la que se ha adaptado un MAO  $(20 \times 20)$ -dimensional (véase la Figura 6.27). Este experimento pretende comparar las imágenes producidas por el algoritmo con las generadas por otros métodos tradicionales de visualización de MAO. Por ejemplo, los basados en matrices de distancias (véanse las Figuras 5.2, 5.4, 5.8 (a) y 5.9), histogramas de datos (véase la Figura 5.12), planos de componentes (véanse las Figuras 5.17 y 5.20) y otras técnicas descritas previamente (véanse las Figuras 5.22, 5.23 (a), 5.26, y 6.46).

La Figura 6.56 muestra el proceso de contracción de neuronas para los instantes  $t = 50$ ,  $100$  y  $200$  en (a), (b) y (c), respectivamente. La imagen (d) ilustra el encadenado con la proyección final (c). El núcleo de vecindad utilizado es el descrito en (6.18) y los parámetros usados corresponden a los “estándar”, descritos en la Sección 6.3.3.1. En las proyecciones generadas es posible observar como las neuronas pertenecientes a cada *cluster* convergen hacia unos centros para formar grupos más densos a medida que se suceden las iteraciones del algoritmo. Por otro

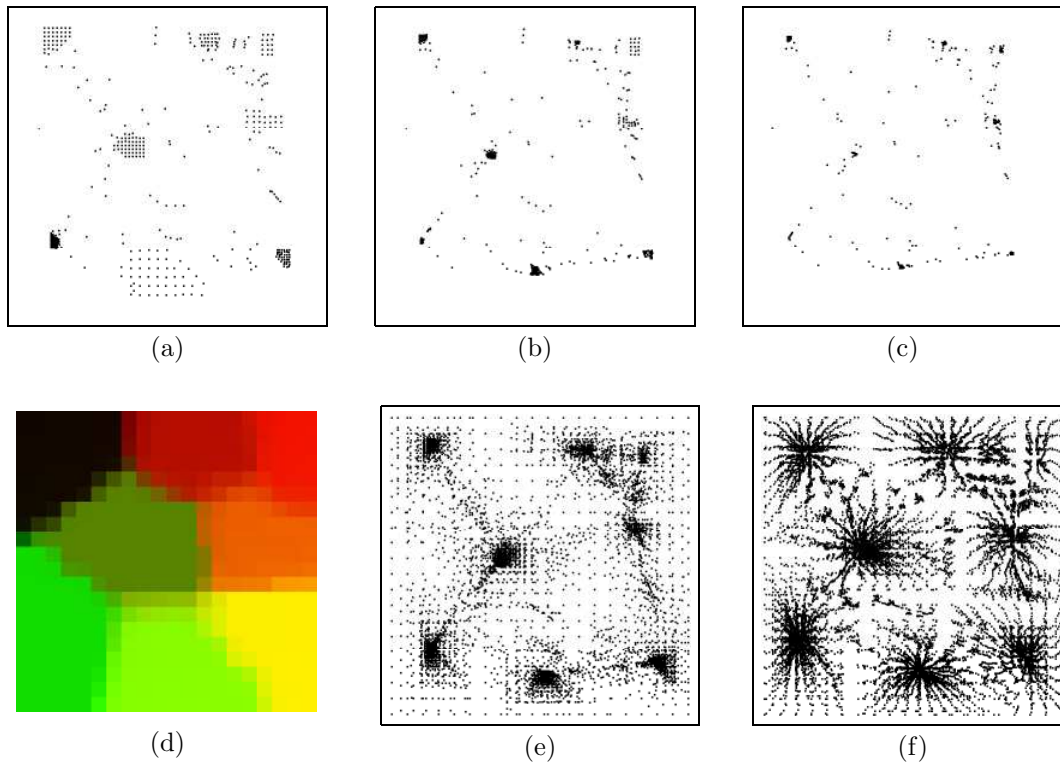


Figura 6.56: Proceso de contracción del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos. Las proyecciones corresponden a los instantes  $t = 50, 100$  y  $200$ , en (a), (b) y (c), respectivamente. La imagen (d) ilustra el encadenado de la proyección final (c) con la original ( $t = 0$ ) del MAO mediante un código de colores lineal. Las imágenes (e) y (f) muestran dos vistas en planta del dendograma que genera el AAN para  $T = 200$  y  $500$  iteraciones, respectivamente.

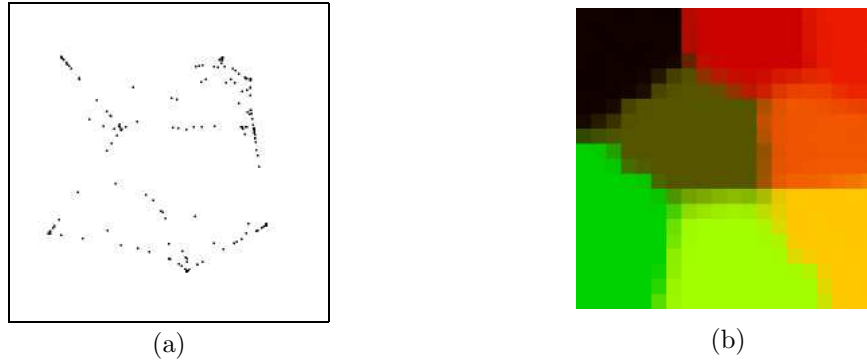


Figura 6.57: Proyección final del método de Himberg (a) y encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por ocho *clusters* disjuntos.

lado, dado el núcleo de vecindad utilizado, en las primeras iteraciones del algoritmo es posible apreciar la estructura regular de las neuronas que representan a los *clusters*. La proyección final (c) es encadenada con la original ( $t = 0$ ) mediante un código de colores lineal para producir una imagen en la que la estructura de *clusters* es representada adecuadamente (d). Nótese la similitud entre la imagen obtenida y las obtenidas por los métodos de visualización citados previamente. En la imagen aparecen los ocho *clusters* claramente, donde las diferencias entre sus colores dan una idea sobre la distancia real que los separa. En cuanto a los bordes entre ellos, su detección es ligeramente más difícil que en el experimento anterior. Esto es debido a que el contraste entre los *clusters* vecinos es menor, dado que el número de éstos ha aumentado. Por esta razón, los métodos de contracción que generan imágenes coloreadas mediante el proceso de encadenado pueden presentar dificultades cuando las distribuciones están compuestas por un número elevado de *clusters*. Las últimas imágenes corresponden a vistas en planta del árbol jerárquico de clasificación o dendrograma que genera el proceso de contracción. Ilustran todas las proyecciones de las neuronas en cada iteración del AAN, es decir, los niveles del dendrograma que corresponden a cada iteración del algoritmo. En (e) los parámetros coinciden con los “estándar”, mientras que en (f) el número de iteraciones es elevado hasta  $T = 500$  y la tasa de aproximación inicial es reducida a  $\lambda(0) = 0,05$  para suavizar la visualización. En general, este tipo de imagen resulta más ruidosa y difícil de interpretar que la visualización de un sólo nivel del dendrograma (proyección generada en una determinada iteración del algoritmo).

En cuanto a la relación con el método de Himberg, la Figura 6.57 muestra la proyección final de este método (con  $\theta = 0,25$  y  $r = 20$ ) y la imagen coloreada tras realizar el encadenado. Obsérvese la similitud entre estas imágenes y las generadas por el AAN, donde la detección de los bordes entre los *clusters* de la esquina superior-derecha del MAO empieza a resultar difícil.

#### 6.3.3.4. Cuatro *clusters* disjuntos de diferente tamaño e igual forma

En la práctica, la diferencia principal entre los métodos de contracción de neuronas y las matrices de distancias reside en el uso del color para representar los diferentes *clusters* y sus bordes. El uso del color es ventajoso ya que permite identificar cada *cluster* con un color determinado, lo cual aporta numerosas ventajas (véase la Sección 6.3.1). Los bordes entre los *clusters* son identificados mediante las diferencias entre los colores asociados a neuronas adyacentes en el mapa. Estas diferencias pueden resultar difíciles de detectar si existen *clusters* muy próximos en el espacio de los datos o si la distribución de datos está compuesta de un número elevado de *clusters*. Nótese que el número de colores de un código de colores que un analista puede diferenciar es limitado, lo cual limita asimismo el número de *clusters* que puede percibir en una sola imagen.

Por otro lado, si la distancia en el espacio de los datos entre dos *clusters* adyacentes en el MAO es pequeña, las matrices de distancias también van a presentar dificultades a la hora de ilustrar el borde entre dichos *clusters*. Entre las soluciones a éste problema cabe la posibilidad de realizar una transformación del histograma de la imagen, para incrementar el contraste en las zonas donde existen bordes. Otra solución consiste en emplear el método de superposición de imágenes (véase la Sección 6.2), el cual incorpora información del histograma de datos para facilitar la visualización de bordes.

En el siguiente experimento un MAO ( $20 \times 20$ )-dimensional es adaptado a la distribución compuesta de cuatro *clusters* disjuntos de diferente tamaño e igual forma, véase la Figura 6.28. La dificultad en este ejemplo radica en la pequeña separación entre los pares de *clusters* centrados en  $X_1 = -10$  y en  $X_1 = 10$  (véase la Figura 4.12), que complica la discriminación de los cuatro *clusters* y la elección de los parámetros de los métodos de contracción. En este ejemplo es necesario reducir la fuerza de atracción del AAN, ya que de lo contrario las neuronas de cada par de *clusters* laterales convergirían rápidamente hacia un solo punto en el espacio de observación. En este caso, tanto la proyección como

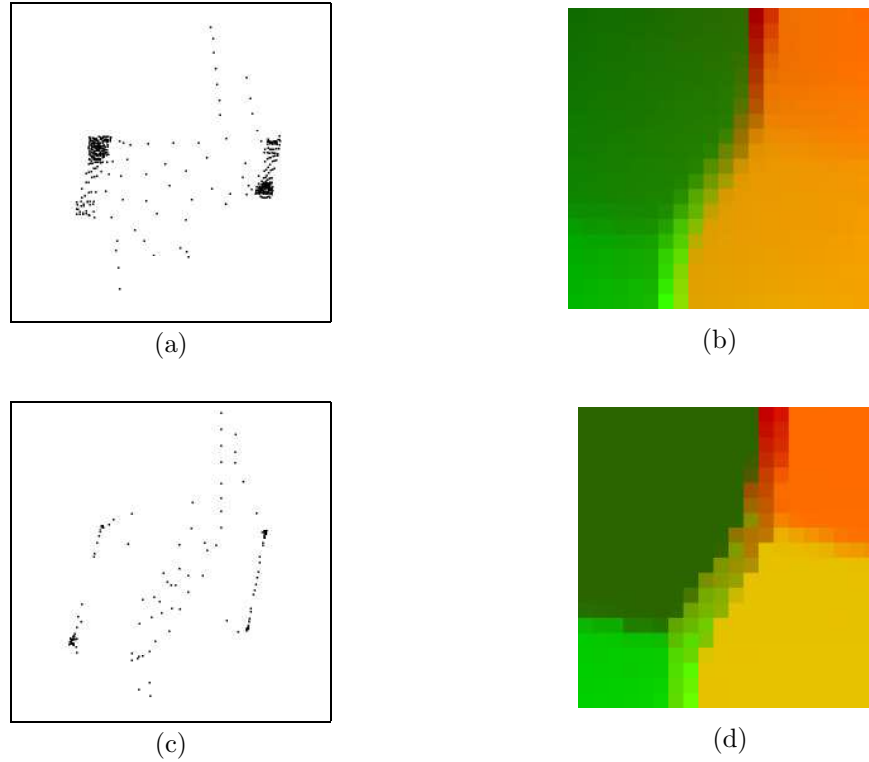


Figura 6.58: Proyecciones generadas con el AAN y encadenado mediante un código de colores lineal para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma. En (a) se emplea el núcleo de vecindad descrito en (6.20), mientras que en (c) se usa el descrito en (6.18). Los correspondientes encadenados de (a) y (c) con la proyección original del MAO son ilustrados en (b) y (d), respectivamente.

la imagen coloreada sólo mostrarían dos *clusters*, como la U-matrix ilustrada en la Figura 6.45 (a). Esto también aparece en la proyección de Himberg si el parámetro  $\theta$  es relativamente elevado, por lo que es necesario escogerlo bajo. En definitiva, la dificultad de este problema es la misma que afecta a todos los algoritmos de *clustering* cuando no es posible conocer *a priori* las características de la distribución a analizar (generalmente el número de *clusters* no es conocido, ni la altura por la que debe ser cortado un dendrograma, etc.). En este caso, el árbol jerárquico suave de clasificación debe ser cortado por un nivel bajo para poder separar los cuatro *clusters*. En general, el analista debe generar varias proyecciones con diferentes parámetros para observar la estructura de *clusters* obtenidos en diferentes niveles del árbol jerárquico o dendrograma.

La Figura 6.58 muestra dos proyecciones finales generadas con el AAN y el correspondiente encadenado mediante un código de colores lineal para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters*

disjuntos de diferente tamaño e igual forma.

En la primera proyección (a), se ha utilizado el núcleo de vecindad descrito en (6.20) y el conjunto de parámetros “estándar” descritos en la Sección 6.3.3.1, a excepción de la tasa de aproximación inicial cuyo valor es reducido a  $\lambda(0) = 0,1$ . Esto es necesario para evitar que las neuronas se concentren alrededor de dos puntos, cada uno en un lateral del MAO, lo cual indicaría la presencia de únicamente dos *clusters*. Con esta tasa de aproximación baja es posible obtener una imagen que ilustra adecuadamente la distribución de los vectores de referencia (refleja adecuadamente las distancias entre los *clusters*, la densidad de neuronas en cada *cluster* y las neuronas interpolantes). Sin embargo, la separación entre los *clusters* laterales (centrados en  $X1 = -10$  y en  $X1 = 10$ ) no es manifiesta, lo cual también queda reflejado en la imagen coloreada basada en el encadenado (b), donde los bordes entre los *clusters* no aparecen nítidamente. En este caso, el algoritmo tiene problemas para conseguir acercar las neuronas de cada *cluster* a cuatro lugares diferentes en el MAO, principalmente por el tipo de núcleo de vecindad utilizado. Como éste está definido en el espacio de los datos y en el de observación, una neurona no sólo atrae a neuronas en función de la distancia entre sus vectores de referencia, sino que también atrae con más fuerza a las neuronas que se hallan próximas a ella en el propio mapa. Por tanto, existe una mayor fuerza de atracción entre *clusters* cercanos en el espacio de los datos y en el mapa. De esta manera, con una tasa de aproximación más elevada los pares de *clusters* laterales son fusionados rápidamente en la proyección, indicando la presencia de únicamente dos *clusters*.

Por otro lado, en (c) se utiliza el núcleo de vecindad descrito en (6.18) y su conjunto de parámetros “estándar” asociado, a excepción del radio inicial del núcleo de vecindad que pasa a ser el 10 % ( $\tau = 0,1$ ) del diámetro del conjunto de vectores de referencia,  $\sigma(0) = 0,1 \cdot \mathcal{D}$ . El resultado es una proyección en la que los *clusters* se encuentran más separados que en (a), mientras que las neuronas asociadas a éstos forman grupos densos en torno a cuatro puntos en el espacio de observación. De esta manera, es posible identificar claramente los cuatro *clusters* tanto en la proyección como en la imagen fruto del encadenado basado en un código de colores lineal (d). En esta segunda imagen cada *cluster* aparece con un color diferente y los bordes entre éstos, formados por neuronas interpolantes (que también pueden ser detectadas con facilidad en la proyección), son claramente visibles. Por otro lado, aunque parezca que la información relativa a la densidad

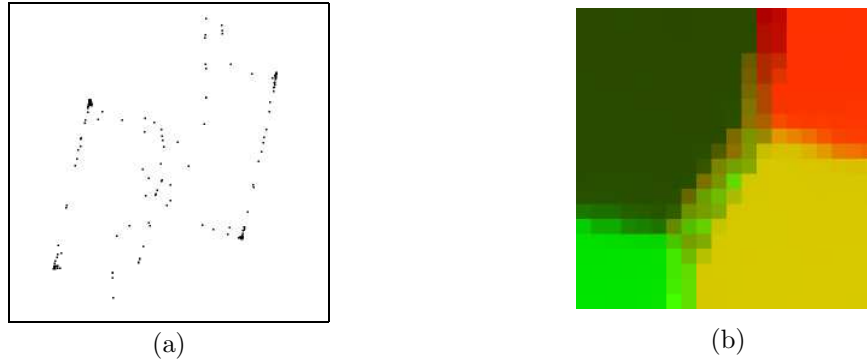


Figura 6.59: Proyección generada con el método de contracción de Himberg (a) y encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de diferente tamaño e igual forma. La proyección ha sido calculada en cinco iteraciones con  $\theta = 0,25$ .

de neuronas en cada *cluster* desaparece en la proyección, ésta puede ser deducida a partir del tamaño de cada *cluster* en la imagen coloreada (generalmente la densidad de datos de un *cluster* es directamente proporcional al tamaño de éste en el MAO). El algoritmo es capaz de revelar adecuadamente la estructura de *clusters* en la imagen coloreada, a pesar de que las distancias relativas entre los *clusters* en la proyección no coinciden con las originales en el espacio de los datos.

El método de Himberg también puede producir descripciones adecuadas de la distribución de datos. La Figura 6.59 ilustra la proyección generada con  $r = 5$  y  $\theta = 0,25$  (a), y la imagen coloreada fruto del encadenado mediante un código de colores lineal (b). Las imágenes son muy similares a las obtenidas en la Figura 6.58 (c) y (d), ya que la fuerza de atracción que implementa el método sólo depende de las distancias entre los vectores de referencia de las neuronas (como el núcleo de vecindad descrito en (6.18)).

#### 6.3.3.5. Cuatro *clusters* de igual tamaño y diferente forma

La elección del conjunto de parámetros a la hora de ejecutar un algoritmo de contracción de neuronas influye principalmente en el grado de atracción de éstas. La variación de los parámetros permite generar diferentes proyecciones, obtenidas en estados concretos de los procesos de contracción (niveles de los árboles jerárquicos de clasificación). En la práctica, el analista debe variar los parámetros para visualizar diversas proyecciones, hasta obtener una idea general de las

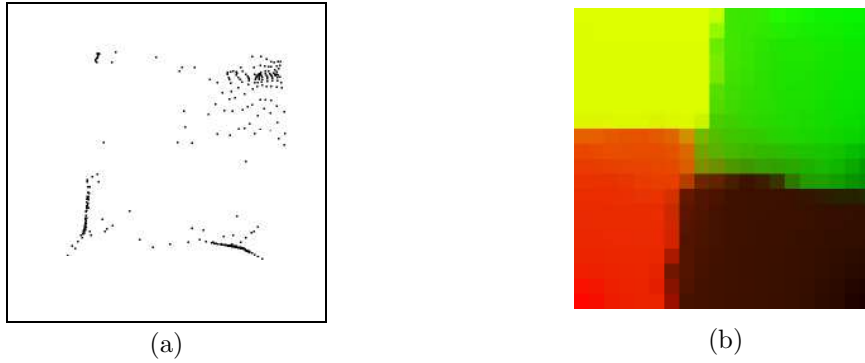


Figura 6.60: Proyección generada con el AAN (a) y encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.



Figura 6.61: Matriz de distancias (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

características de la distribución y posiblemente descubrir estructuras inesperadas en los datos. Es cierto que puede existir un conjunto de parámetros “óptimo”, con el que la proyección generada describa la distribución de datos lo mejor posible según algún criterio (por ejemplo, los relacionados con el EMD, como la preservación de las distancias originales en el espacio de observación). No obstante, la necesidad de obtener una imagen precisa de la distribución de datos no suele ser crítica, especialmente si se desea analizar la imagen coloreada tras realizar un encadenado basado en un código de colores. En otras palabras, proyecciones obtenidas con diferentes parámetros suelen producir imágenes coloreadas similares.

La Figura 6.60 muestra la proyección del AAN (a) y el encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional (véase



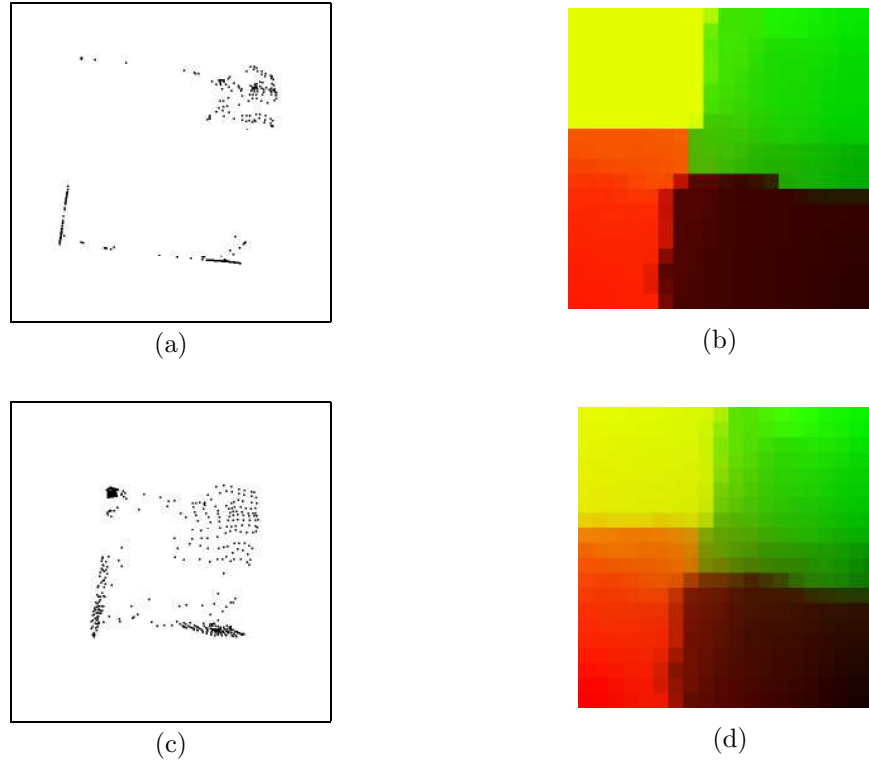


Figura 6.62: Proyecciones generadas con el método de contracción de Himberg y encadenados basados en un código de colores lineal para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma. En (a) la proyección ha sido calculada en  $r = 20$  iteraciones con  $\theta = 100$ . En (c)  $r = 1$  y  $\theta = 5000$ . El correspondiente encadenado de (a) y (c) con la proyección original del MAO es ilustrado en (b) y (d), respectivamente.

la Figura 6.29) adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma. En este experimento se ha utilizado el núcleo de vecindad descrito en (6.20) y el conjunto de parámetros coincide exactamente con el “estándar” (véase la Sección 6.3.3.1). Aunque la proyección no produce una descripción perfecta de la distribución de datos, la imagen coloreada refleja adecuadamente como el MAO ha sido adaptado a los datos. Ésta revela claramente la existencia de cuatro *clusters*, los cuales lógicamente aparecen con un tamaño similar en el MAO, y donde incluso es posible intuir su forma. La imagen es similar a las generadas por métodos tradicionales de visualización, véase la Figura 6.61.

Por otro lado, el método de Himberg puede producir una proyección similar a la ilustrada en la Figura 6.60 (a), además de otra más precisa de la distribución de datos o vectores de referencia. La Figura 6.62 (a) muestra la proyección del

método tras  $r = 20$  iteraciones con  $\theta = 100$ . En (c),  $\theta$  es aumentado hasta 5000, por lo que sólo ha sido necesario ejecutar una iteración del algoritmo para generar la proyección, la cual describe los datos de forma más precisa. Lo más interesante del experimento resulta ser la similitud entre las imágenes coloreadas (b) y (d) de esta figura y la (b) de la Figura 6.60, a pesar de haber sido generadas con algoritmos distintos y parámetros diferentes.

En general, las imágenes coloreadas suelen ser bastante robustas a la configuración final de las neuronas en el espacio de observación tras ejecutar un algoritmo de contracción. Por esta razón, es posible emplear métodos rápidos de contracción como el AAN, ya que la pérdida de precisión no suele ser crítica a la hora de analizar tanto las proyecciones como las imágenes generadas a partir de un encadenado con un código de colores.

#### 6.3.3.6. Un segmento

En el siguiente experimento el objetivo consiste en analizar el comportamiento del algoritmo cuando la dimensión intrínseca de los datos es uno. La distribución escogida es un segmento definido en un espacio 8-dimensional. En este caso, un MAO bidimensional va a tener problemas para adaptarse de forma ordenada a la distribución, y no va a ser posible desvelar una estructura de *clusters*, ya que ésta no aparece en la distribución. De esta manera, las imágenes producidas por los métodos diseñados para observar la estructura de *clusters* van a ser difíciles de interpretar, incluyendo las producidas por un encadenado basado en un código de colores. Con respecto a la forma del MAO, en este caso parece más razonable utilizar uno rectangular que uno cuadrado, ya que existe una clara dirección de mayor varianza en los datos. Sin embargo, se ha optado por no variar la forma de éstos (cuadrada en la mayoría de ejemplos presentados en esta tesis), principalmente porque un MAO bidimensional cuadrado puede adaptarse a un segmento de dos formas distintas que resultan interesantes de observar con el AAN y otros métodos:

1. El segmento se ubica u organiza en el MAO según una de sus diagonales. Es decir, los extremos del segmento aparecen en esquinas opuestas del mapa.
2. El segmento recorre el MAO desde una esquina determinada hasta otra contigua a ésta, apareciendo una figura en forma de “U”.

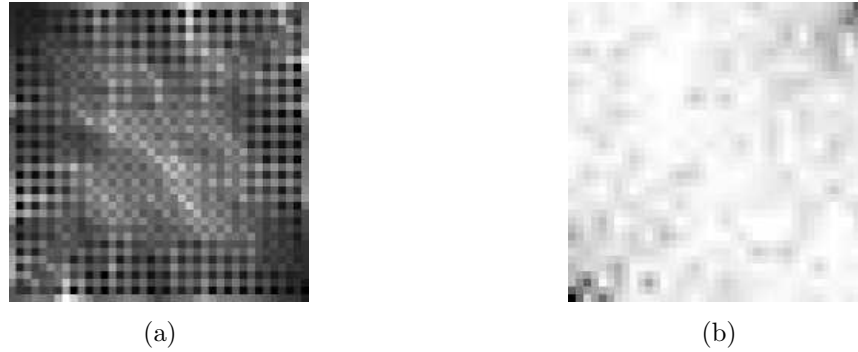


Figura 6.63: U-matrix (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa según su diagonal secundaria.

Los resultados relacionados con ambas formas de adaptación de un MAO  $(20 \times 20)$ -dimensional a un segmento (véase la Figura 6.36) son presentados a continuación.

En el primer caso el segmento queda organizado según la diagonal secundaria ( $45^\circ$ ) del MAO, donde un extremo del segmento está ubicado en la esquina inferior-izquierda y el otro en la superior-derecha. Ni la U-matrix ni el histograma de datos, representados en la Figura 6.63 pueden aportar una descripción adecuada de la distribución de datos. Las imágenes son muy difíciles de interpretar ya que no aparece una estructura de *clusters*.

La Figura 6.64 muestra el proceso de contracción producido por el AAN, donde el núcleo de vecindad utilizado corresponde al descrito en (6.18) y los parámetros son los “estándar” descritos en la Sección 6.3.3.1. Las imágenes corresponden a los instantes de tiempo  $t = 5, 10, 20, 40, 100$  y  $200$ , en (a-f), respectivamente. Queda claro en el ejemplo que la proyección va tomando progresivamente la forma de un segmento. Resulta igualmente interesante ver el proceso dinámico, donde parece que las neuronas empiezan a “enrollarse” alrededor de un segmento imaginario en el mapa, donde es posible observar que las neuronas de una misma diagonal paralela a la principal ( $135^\circ$ ) representan datos muy próximos en el segmento.

Este fenómeno también puede ser detectado en las imágenes coloreadas fruto del encadenado con un código de colores. La Figura 6.65 muestra el encadenado con un código de colores para la primera (a) y última (f) imagen de la Figura 6.64. Las imágenes son difíciles de interpretar por sí solas ya que no aparece una estructura de *clusters*. Sin embargo, las neuronas localizadas en franjas diagonales ( $135^\circ$ ) toman un color muy parecido, indicando que en realidad representan datos muy próximos en el segmento.

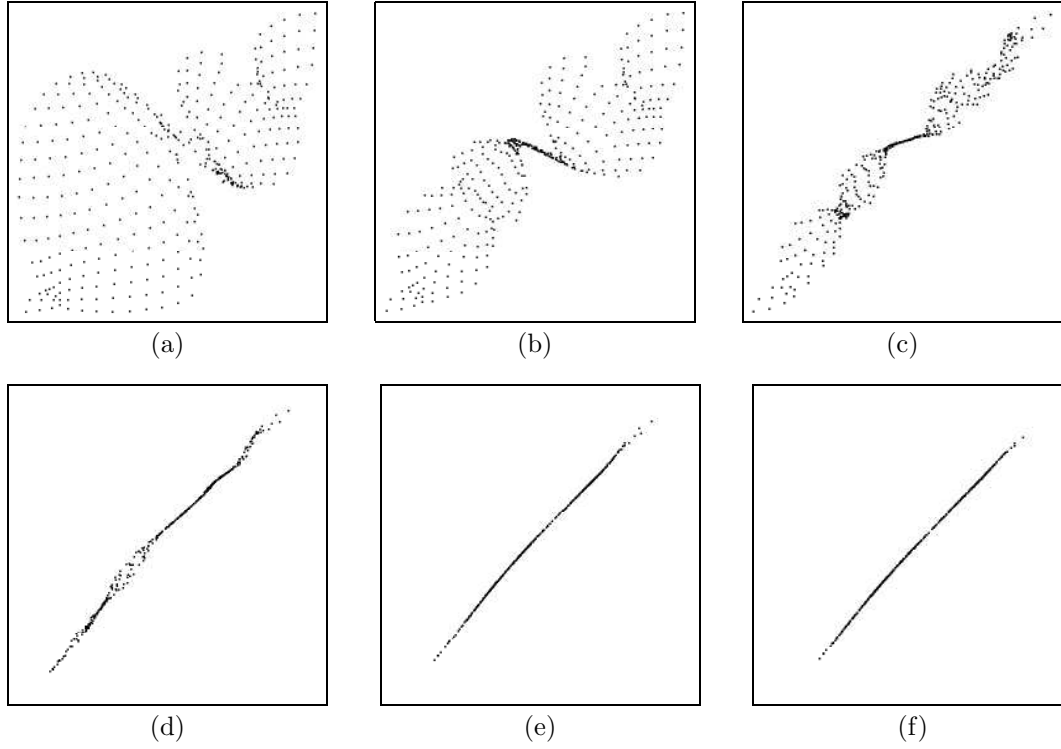


Figura 6.64: Proceso de contracción del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa según su diagonal secundaria. Las imágenes corresponden a los instantes de tiempo  $t = 5, 10, 20, 40, 100$  y  $200$ , en (a-f), respectivamente.

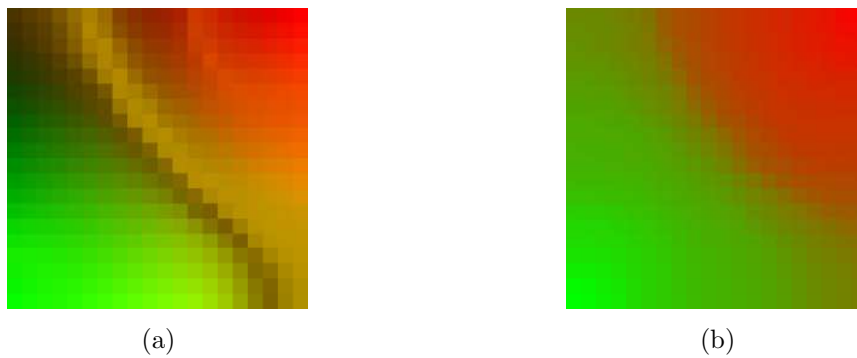


Figura 6.65: Encadenado con un código de colores lineal para la primera (a) y última (f) imagen de la Figura 6.64.

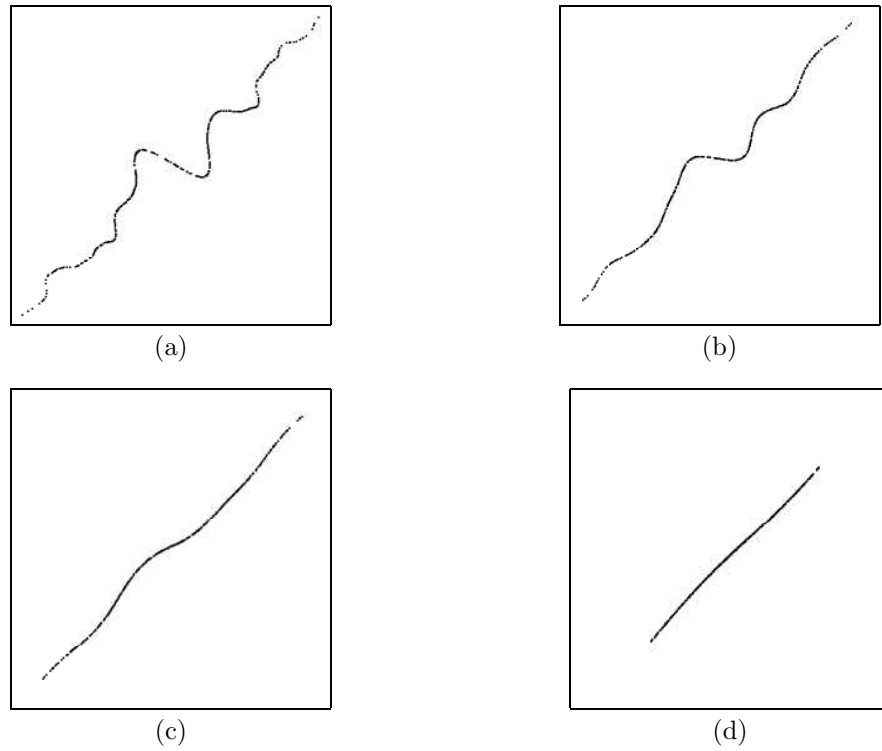


Figura 6.66: Proceso de contracción del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa según su diagonal secundaria. Las imágenes corresponden a las proyecciones obtenidas con  $\theta = 0,25$  en las iteraciones  $r = 1, 5, 20$  y  $100$ , en (a-d), respectivamente.

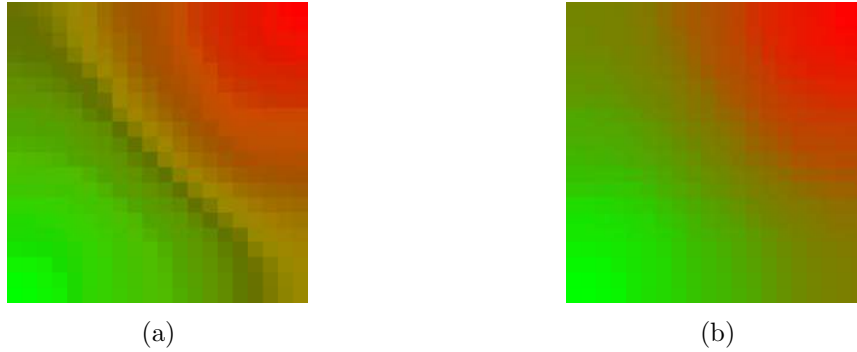


Figura 6.67: Encadenado con un código de colores lineal para la primera (a) y última (d) imagen de la Figura 6.66.

El método de contracción de Himberg también revela adecuadamente la forma lineal de los datos. Los resultados que genera son similares a los obtenidos con el AAN, aunque el proceso de atracción de las neuronas es muy diferente. La Figura 6.66 muestra las proyecciones que genera el método para  $\theta = 0,25$  y  $r = 1, 5, 20$  y  $100$ , ilustradas en (a-d), respectivamente. En la primera iteración es posible empezar a intuir la forma de los datos, al aparecer una curva continua organizada a lo largo de la diagonal secundaria del MAO. A medida que transcurren las iteraciones la curva se aproxima cada vez más a un segmento, donde no existe una diferencia significativa entre la proyección final de este método y la del AAN.

Por último, las imágenes (a) y (b) de la Figura 6.67 muestran el encadenado con un código de colores lineal de las proyecciones ilustradas en las imágenes (a) y (d), respectivamente, de la Figura 6.66. Al igual que en la Figura 6.65, las imágenes indican que las neuronas están organizadas en franjas paralelas a la diagonal principal.

Un MAO también puede adaptarse a un segmento de forma que éste quede organizado en el mapa en forma de “U” (el segmento queda representado desde una esquina hasta otra contigua a ésta en el mapa). Esto puede ser detectado en la U-matrix, ilustrada en la Figura 6.68 (a), donde aparece una clase curvada en forma de “ $\cap$ ” con un borde de distancias elevadas en la zona inferior-central. Por el contrario, el histograma de datos (b) no aporta información útil al no desvelar la estructura de los datos. Al igual que en el caso anterior, el MAO utilizado es  $(20 \times 20)$ -dimensional.

La Figura 6.69 muestra el proceso de contracción producido por el AAN para este nuevo MAO, donde el núcleo de vecindad utilizado corresponde al descrito

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

---



Figura 6.68: U-matrix (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, donde sus neuronas aparecen organizadas en forma de “C”.

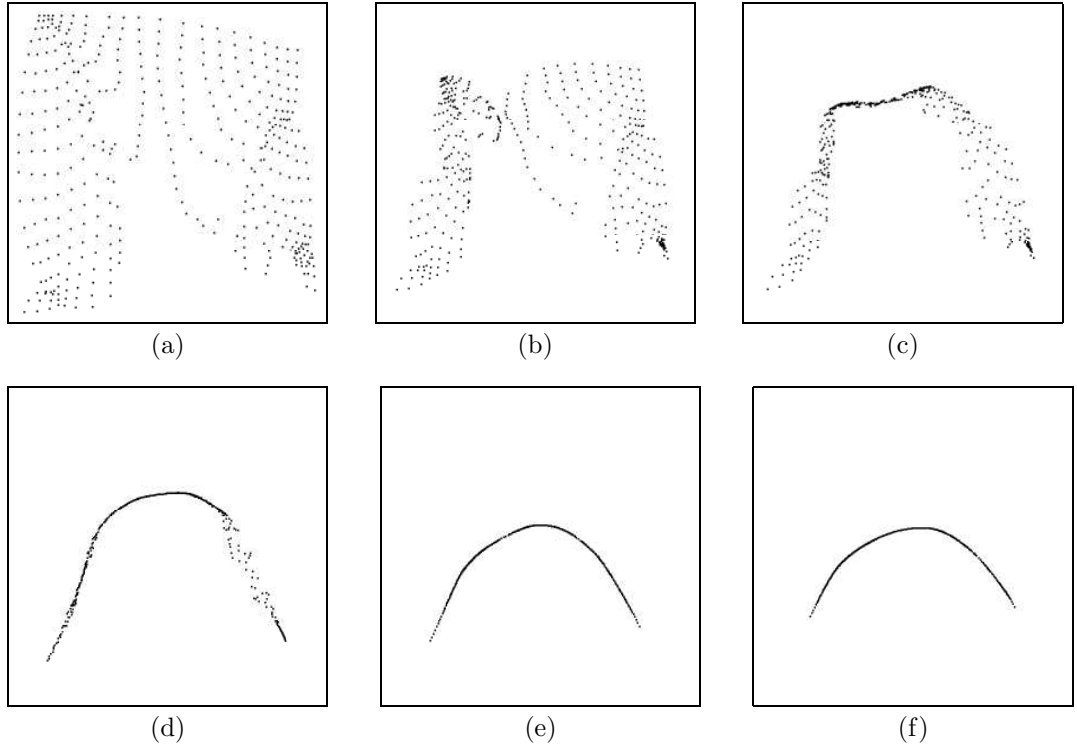


Figura 6.69: Proceso de contracción del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa en forma de “C”. Las imágenes corresponden a los instantes de tiempo  $t = 5, 10, 20, 40, 100$  y  $200$ , en (a-f), respectivamente.

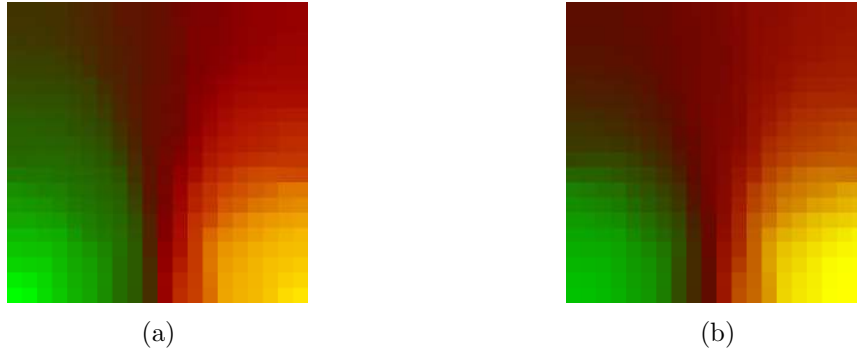


Figura 6.70: Encadenado basado en un código de colores lineal para la última proyección (f) de la Figura 6.69 y para la última proyección (d) de la Figura 6.71.

en (6.18) y los parámetros son los “estándar”. Las imágenes corresponden a los instantes de tiempo  $t = 5, 10, 20, 40, 100$  y  $200$ , en (a-f), respectivamente. Tal y como ocurría en el ejemplo de la Figura 6.64, en el proceso dinámico las neuronas se aproximan hasta formar una curva continua de dimensión intrínseca uno, donde parece que éstas se “enrollan” alrededor de una figura en forma de “ $\cap$ ”. Aproximadamente, a partir de las primeras 40 iteraciones el método es capaz de representar adecuadamente cómo el MAO se ha adaptado al segmento. Posteriormente, a medida que se suceden las iteraciones del algoritmo la curva tiende a parecerse a un segmento. De hecho, es necesario elevar la tasa inicial de aproximación (por ejemplo,  $\lambda(0) = 2,5$ ) para que al final del proceso aparezca una curva prácticamente recta.

El encadenado de la proyección final del algoritmo tras 200 iteraciones con la original del MAO mediante un código de colores lineal produce la imagen ilustrada en la Figura 6.70 (a). La imagen (b) muestra el encadenado con la proyección creada por el método de Himberg tras realizar 200 iteraciones con  $\theta = 1$  (véase la Figura 6.71). Ambas imágenes son muy parecidas y resultan difíciles de interpretar dada la dimensión intrínseca del segmento. Sin embargo, con la ayuda de las proyecciones es posible determinar los extremos del segmento, el borde entre éstos y en definitiva cómo queda representado el segmento por las neuronas en el mapa.

La Figura 6.71 muestra las proyecciones generadas por el método de Himberg para  $\theta = 1$  y  $r = 5, 50, 100$  y  $200$  en (a-d), respectivamente. El encadenado de la última proyección es ilustrado en la Figura 6.70 (b). Los resultados son similares a los obtenidos mediante el AAN (no existe una diferencia significativa entre



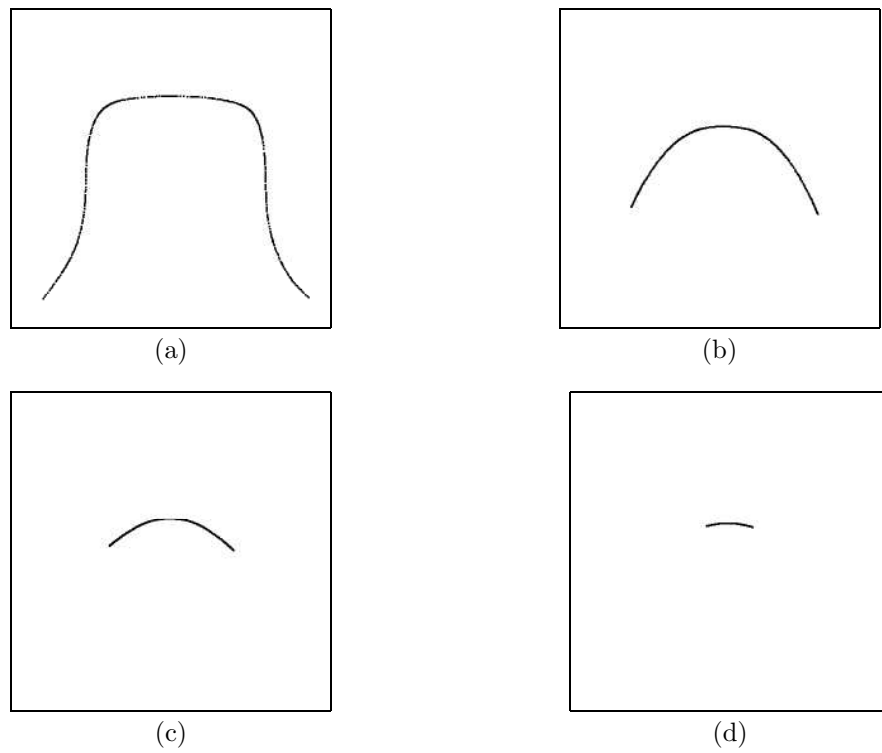


Figura 6.71: Proceso de contracción del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa en forma de “ $\cap$ ”. Las imágenes corresponden a las proyecciones obtenidas con  $\theta = 1$  en las iteraciones  $r = 5, 50, 100$  y  $200$ , en (a-d), respectivamente.

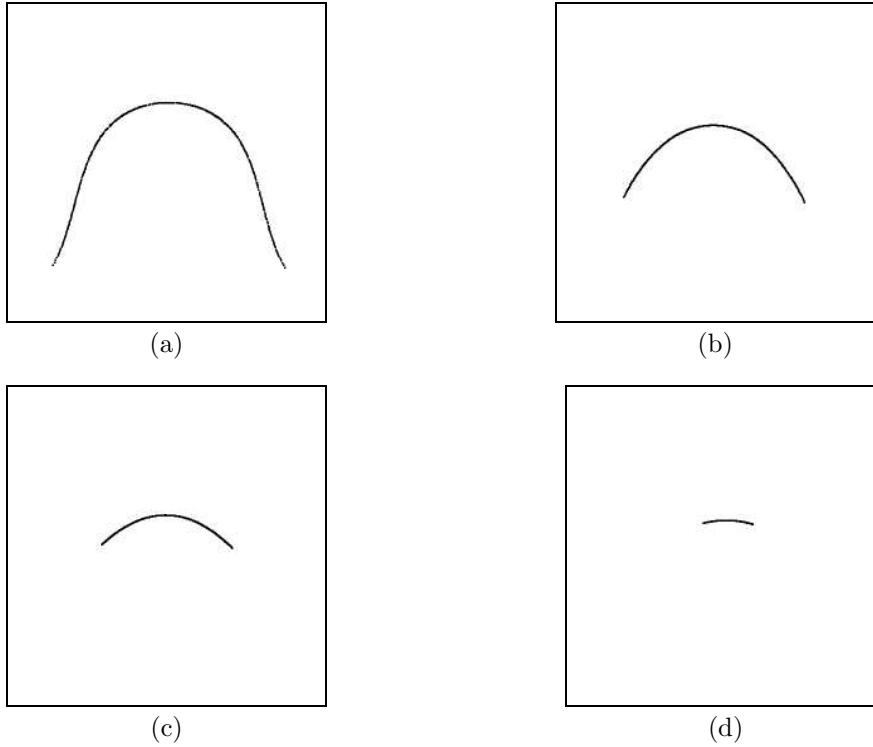


Figura 6.72: Proceso de contracción del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a un segmento, organizado en el mapa en forma de “ $\cap$ ”. Las imágenes corresponden a las proyecciones obtenidas con  $\theta = 20$  en las iteraciones  $r = 1, 3, 5$  y  $10$ , en (a-d), respectivamente.

las proyecciones finales de ambos métodos), a pesar de que en el ejemplo de la Figura 6.69 la baja tasa de aproximación no es suficiente como para ilustrar una curva recta al final del proceso. En este caso, el método de contracción de Himberg también revela adecuadamente la adaptación del MAO al segmento a partir de las primeras iteraciones. El proceso dinámico permite visualizar como la proyección avanza desde la forma con la que el MAO representa al segmento (curva con forma de “ $\cap$ ”), hasta su forma original (lineal). Por otro lado, en este experimento queda patente la necesidad de reescalar las proyecciones (independientemente del método de contracción utilizado), ya que una fuerza de atracción elevada o demasiadas iteraciones puede provocar una pérdida de resolución en las imágenes debida a una aglomeración densa de las neuronas.

Este experimento también resulta interesante a la hora de analizar la velocidad del método de contracción de Himberg, mediante la relación entre el número de iteraciones  $r$  y el parámetro  $\theta$ . Nótese que teóricamente  $\theta$  no afecta a la velocidad del algoritmo (aunque curiosamente sí en la práctica, véase la Sección 6.3.4), sino

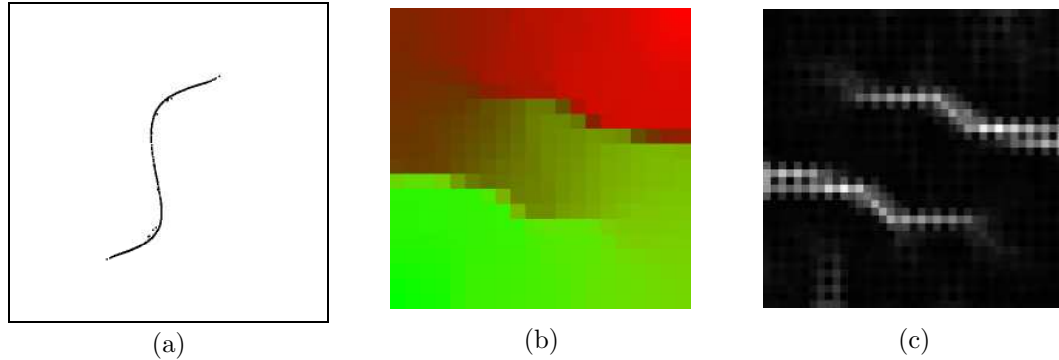


Figura 6.73: Proyección generada con el AAN (a), correspondiente encadenado mediante un código de colores lineal (b) y la U-matrix (c), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un periodo de una senoide. El núcleo de vecindad corresponde al descrito en (6.20), mientras que el conjunto de parámetros es el “estándar”.

al grado de atracción de las neuronas. En el ejemplo anterior es necesario ejecutar un número elevado de iteraciones para conseguir visualizar la estructura lineal de los datos, ya que el parámetro  $\theta = 1$  resulta ser relativamente bajo. Por otro lado, el analista puede estar interesado únicamente en la proyección final. En ese caso, es posible aumentar el parámetro  $\theta$  para elevar la fuerza de atracción entre las neuronas y acelerar el algoritmo disminuyendo el número de iteraciones a ejecutar. De esta manera, es posible obtener prácticamente las mismas proyecciones con un número significativamente más reducido de iteraciones. La Figura 6.72 muestra el mismo ejemplo, donde  $\theta$  ha sido incrementado hasta 20, mientras que el número de iteraciones es reducido a  $r = 1, 3, 5$  y 10, en (a-d), respectivamente. En general, es posible obtener proyecciones que describen adecuadamente los conjuntos de datos con los que se ha entrenado un MAO utilizando una sola iteración ( $r = 1$ ) del método de Himberg, siendo necesario elevar adecuadamente el parámetro  $\theta$ . Por esta razón, la complejidad computacional del método puede ser reducida de  $\mathcal{O}(RM^2)$  a  $\mathcal{O}(M^2)$ , si sólo se desea analizar la proyección final.

### 6.3.3.7. Sinusoide

El siguiente experimento, muy similar al anterior, consiste en analizar los resultados del AAN aplicado a un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un periodo de una senoide, véase la Figura 6.37. La Figura 6.73 muestra la proyección generada con el AAN (a), el correspondiente encadenado mediante un código de colores lineal (b) y la U-matrix (c). En este

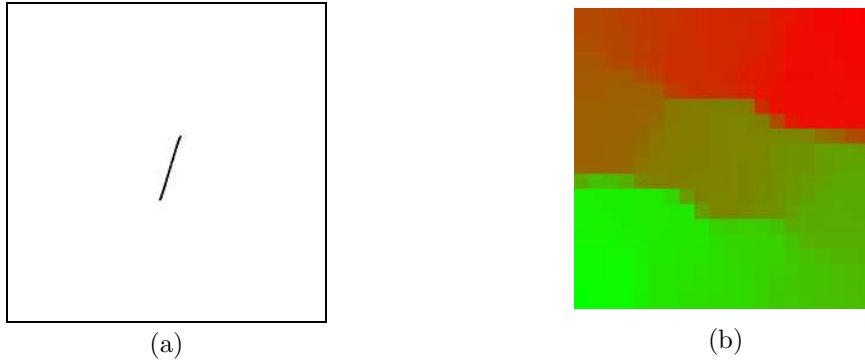


Figura 6.74: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un periodo de una senoide. El núcleo de vecindad corresponde al descrito en (6.18), mientras que el conjunto de parámetros es el “estándar”.

experimento el núcleo de vecindad corresponde al descrito en (6.20) y el conjunto de parámetros coincide exactamente con el “estándar”. Tanto la proyección como la imagen coloreada describen adecuadamente la distribución (nótese la similitud entre esta última y la U-matrix).

Los parámetros escogidos en el ejemplo permiten observar una curva parecida a un periodo de senoide en la proyección. Sin embargo, al emplear más iteraciones o una tasa de aproximación más elevada la curva tiende a aproximarse a un segmento o incluso a un punto (al subir de nivel en el dendograma de clasificación). De hecho, la proyección que genera el algoritmo con el núcleo de vecindad descrito en (6.18) y el conjunto de parámetros “estándar” es muy parecida a un segmento. Ésta es ilustrada en la Figura 6.74 (a), junto con el encadenado basado en un código de colores lineal (b). Esto naturalmente también ocurre cuando el método de Himberg es aplicado con un número de iteraciones  $r$  o un valor de  $\theta$  elevados. Sin embargo, a pesar de que en estos casos la proyección no representa a la distribución de datos con precisión, la imagen coloreada producida a través del encadenado sigue representando una figura sinusoidal, muy parecida a las imágenes (b) y (c) de la Figura 6.73.

Por último, la proyección generada con el método de Himberg para  $\theta = 0,25$  en  $r = 10$  iteraciones es mostrada en la Figura 6.75 (a), la cual es más precisa que la del AAN, al representar más fielmente la forma original de la distribución. En cualquier caso, el correspondiente encadenado basado en un código de colores lineal (b) aporta la misma información que el resto de encadenados generados a partir del AAN, ilustrados en las Figuras 6.73 (b) y 6.74 (b).

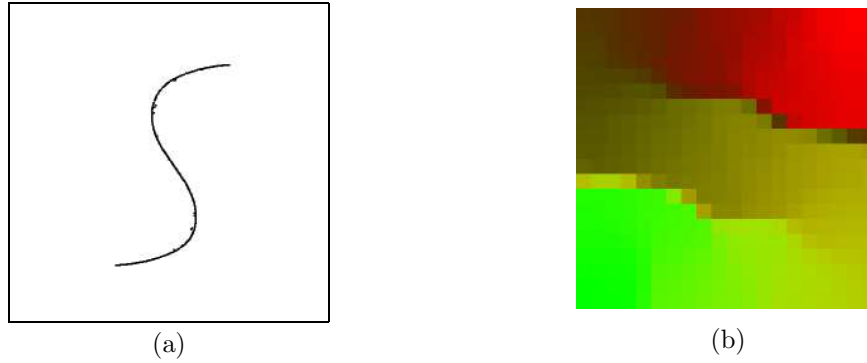


Figura 6.75: Proyección generada con el método de contracción de Himberg (a) y encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un periodo de una senoide. La proyección ha sido calculada en  $r = 10$  iteraciones con  $\theta = 0,25$ .



Figura 6.76: Matriz de distancias (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos segmentos.

#### 6.3.3.8. Dos segmentos

La siguiente distribución está compuesta por dos segmentos de igual longitud definidos en  $\mathbb{R}^8$ . Los métodos tradicionales de visualización de MAO son incapaces de revelar la dimensión intrínseca de los datos, aunque discriminan adecuadamente las dos clases de la distribución. La Figura 6.76 muestra la matriz de distancias (a) y el histograma de datos para un MAO  $(20 \times 20)$ -dimensional adaptado a dicha distribución, véase la Figura 6.77. Ambas imágenes representan las dos clases separadas por un borde claro, mientras que la poca variación de los tonos de gris más oscuros proporciona una pista sobre la uniformidad de los datos en cada clase.

En las imágenes de los métodos tradicionales no es posible detectar si las dos clases representan *clusters*. Para observar la estructura lineal de los datos

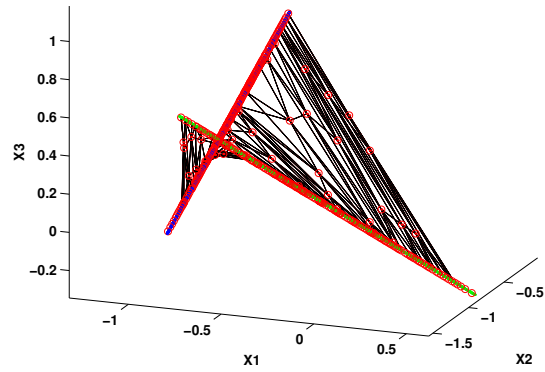


Figura 6.77: MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por dos segmentos.

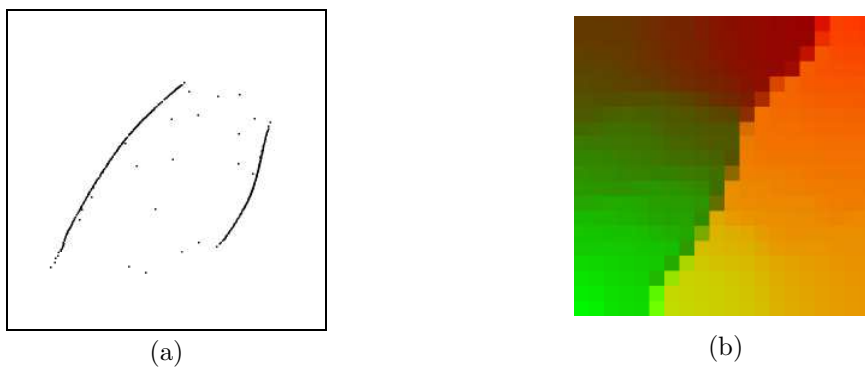


Figura 6.78: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b) para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por dos segmentos.

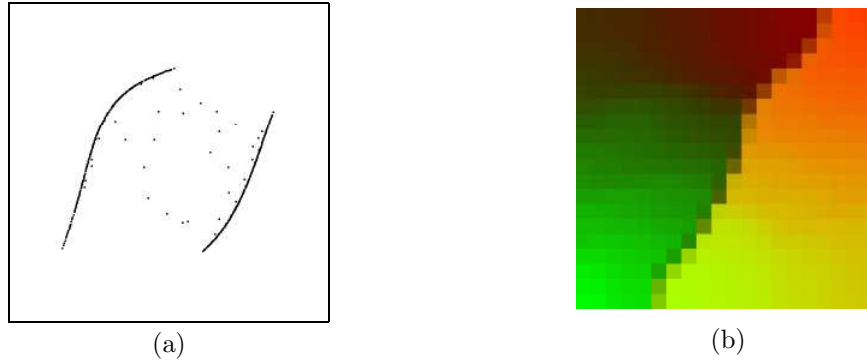


Figura 6.79: Proyección generada con el método de contracción de Himberg con  $\theta = 1$  y  $r = 1$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos segmentos.

es necesario recurrir a otro tipo de métodos como los de contracción. La Figura 6.78 muestra la proyección generada con el AAN para el mismo MAO  $(20 \times 20)$ -dimensional (a), donde es posible observar la dimensión intrínseca de cada clase. La proyección es generada a partir del núcleo de vecindad descrito en (6.20) y el conjunto de parámetros “estándar”. El correspondiente encadenado produce la imagen coloreada ilustrada en (b), la cual consigue discriminar ambas clases mediante las diferencias en los colores de neuronas vecinas. La imagen coloreada es similar a las generadas con los métodos tradicionales, aunque la transición continua de colores en las clases indica que éstas no corresponden a *clusters* compactos sino a estructuras mas “alargadas”.

Por otro lado, el método de contracción Himberg produce prácticamente las mismas proyecciones que el AAN. La Figura 6.79 muestra la proyección generada con el método (a), para  $\theta = 1$  y  $r = 1$ , y el correspondiente encadenado mediante un código de colores lineal (b). Nótese la similitud entre estas imágenes y las ilustradas en la Figura 6.78.

#### 6.3.3.9. Circunferencia

La circunferencia puede ser considerada como una distribución de dimensión intrínseca uno ya que sus datos representan una curva (localmente) y no una superficie. De esta manera, un MAO bidimensional sufre los mismos problemas de desorganización al adaptarse a esta distribución que a segmentos u otras curvas como el periodo de una senoide. Las neuronas situadas en regiones planas del MAO compiten por adaptarse a una curva “unidimensional”, lo cual provoca

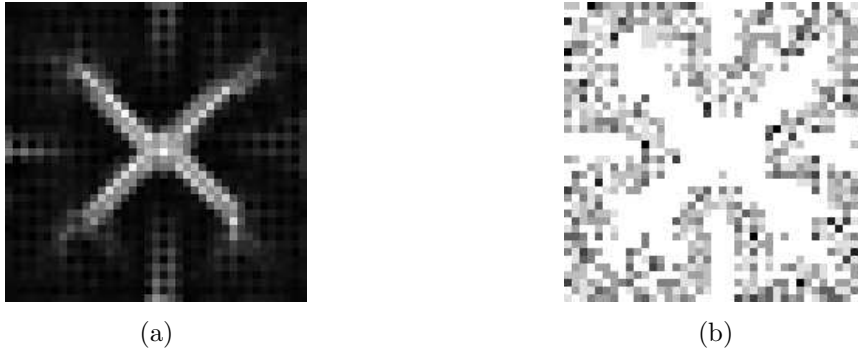


Figura 6.80: U-matrix (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a una circunferencia.

que estas zonas del MAO se “compriman”, distorsionándose para acoplarse a los datos. Afortunadamente, al estar definida en un plano, la circunferencia (al igual que el periodo de una senoide) puede ser modelada adecuadamente por un MAO bidimensional, donde los métodos de visualización pueden revelar algunas características de su forma, como su simetría y estructura cíclica. Casualmente, resulta preferible modelar esta distribución por un MAO cuadrado que por uno rectangular al no existir una dirección de mayor varianza en los datos.

La Figura 6.80 muestra la información que aportan métodos tradicionales de análisis visual de MAO como la U-matrix (a) y el histograma de datos (b), generado a partir de la variante descrita en la Tabla 5.3. El MAO adaptado a una circunferencia es  $(20 \times 20)$ -dimensional, véase la Figura 6.32. Las imágenes muestran como los datos quedan representados por las neuronas del MAO en una figura cíclica simétrica parecida al contorno de una “X”. Como era de esperar, la circunferencia recorre el exterior del MAO, pero tiende hacia el centro a medida que las neuronas se encuentran más alejadas de las esquinas. A pesar de no existir *clusters* en los datos, estas imágenes revelan adecuadamente varias características importantes de la distribución, aunque no son capaces de indicar su dimensión intrínseca.

Los métodos de contracción resultan especialmente indicados para representar datos cuya dimensión intrínseca es inferior a la del propio MAO. El AAN es capaz de crear descripciones adecuadas y precisas de una circunferencia. La Figura 6.81 muestra varias proyecciones del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a una circunferencia, además del encadenado basado en un código de



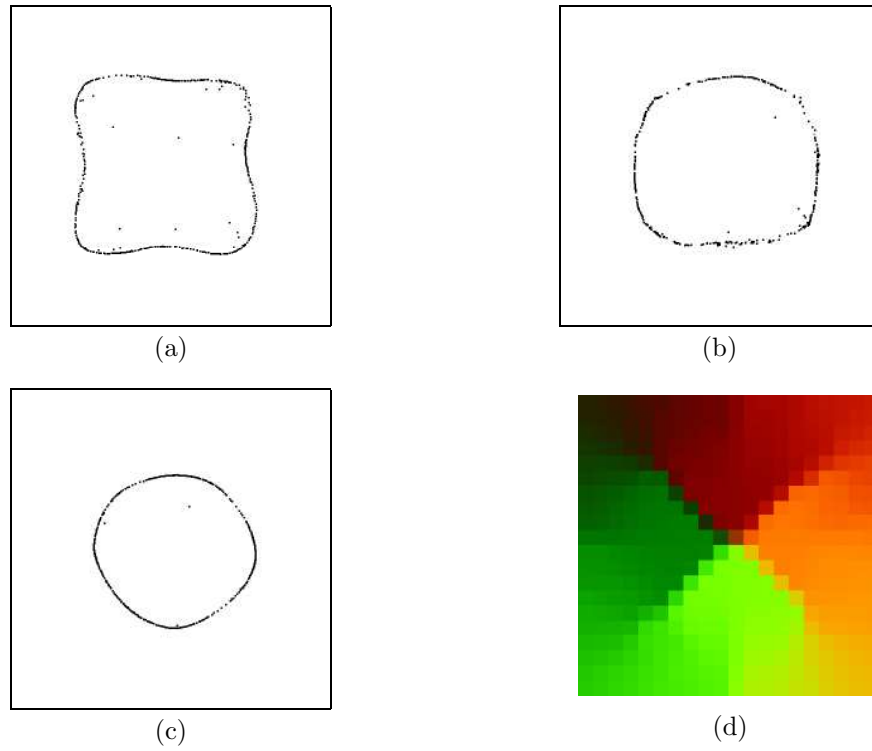


Figura 6.81: Proyecciones del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a una circunferencia y encadenado basado en un código de colores lineal. En (a) y (b) el conjunto de parámetros corresponde al “estándar” para los núcleos de vecindad descritos en (6.20) y (6.18), respectivamente. La proyección ilustrada en (c) utiliza (6.18), con  $T = 1000$  y  $\lambda(0) = 0,5$ , y sirve para generar la imagen coloreada (d) tras realizar un encadenado con la proyección del MAO.

colores lineal con la proyección más precisa. En (a) y (b) el conjunto de parámetros corresponde al “estándar” para los núcleos de vecindad descritos en (6.20) y (6.18), respectivamente. La proyección en (a) es muy similar a la obtenida mediante el MST (véase la Figura 6.32), donde la forma de la curva resultante es similar a la forma en que los datos son representados en el MAO, que puede ser apreciada en la U-matrix y el histograma de datos de la Figura 6.80. La proyección en (b) ilustra una curva más redonda, algo más precisa que la mostrada en (a). La última proyección (c) está generada con el núcleo de vecindad descrito en (6.20), pero el número de iteraciones  $T$  es aumentado hasta 1000, mientras que la tasa de aproximación inicial es reducida a  $\lambda(0) = 0,5$ . De esta manera, la proyección resulta ser más precisa que las anteriores. Finalmente, el encadenado de la proyección original del MAO con la de la imagen (c) genera la imagen coloreada (d). Ésta es similar a la producida por los métodos tradicionales de análisis de MAO y aporta la misma información. La continuidad de la circunferencia (su estructura cíclica) queda manifiesta en la continuidad de los colores a través del contorno en forma de “ $\times$ ”.

Por otro lado, el método de Himberg es capaz de generar descripciones más precisas de la circunferencia. La Figura 6.82 ilustra tres proyecciones del método obtenidas con los parámetros:  $\theta = 0,25$  y  $r = 20$  en (a),  $\theta = 25$  y  $r = 1$  en (b), y  $\theta = 150$  y  $r = 1$  en (c). La primera proyección ilustra la forma en que los datos son representados en el MAO (a). En (b) la imagen es similar a las creadas por el MST y por el propio AAN (véanse las Figuras 6.32 y 6.81). Al subir de nivel en el dendograma creado por el método la curva es transformada hasta presentar una forma bastante precisa de una circunferencia (c). La imagen (d) muestra el encadenado de la proyección del MAO con esta última proyección. La imagen coloreada es muy similar a la presentada en la Figura 6.81 (d).

### 6.3.3.10. Dos eslabones encadenados

En el siguiente experimento la distribución de datos elegida es la formada por dos eslabones encadenados, la cual ha sido analizada previamente con varios métodos de visualización de MAO (véanse las Figuras 5.6, 5.17(b), 5.23 (b), 5.40 (a), 5.44 (a) y 6.30). La adaptación de un MAO bidimensional a esta distribución es ilustrada en la Figura 6.30 con un MAO  $(20 \times 20)$ -dimensional.

La Figura 6.83 muestra la proyección del AAN para dicho MAO  $(20 \times 20)$ -dimensional (a). El correspondiente encadenado es ilustrado en (b). La proyección

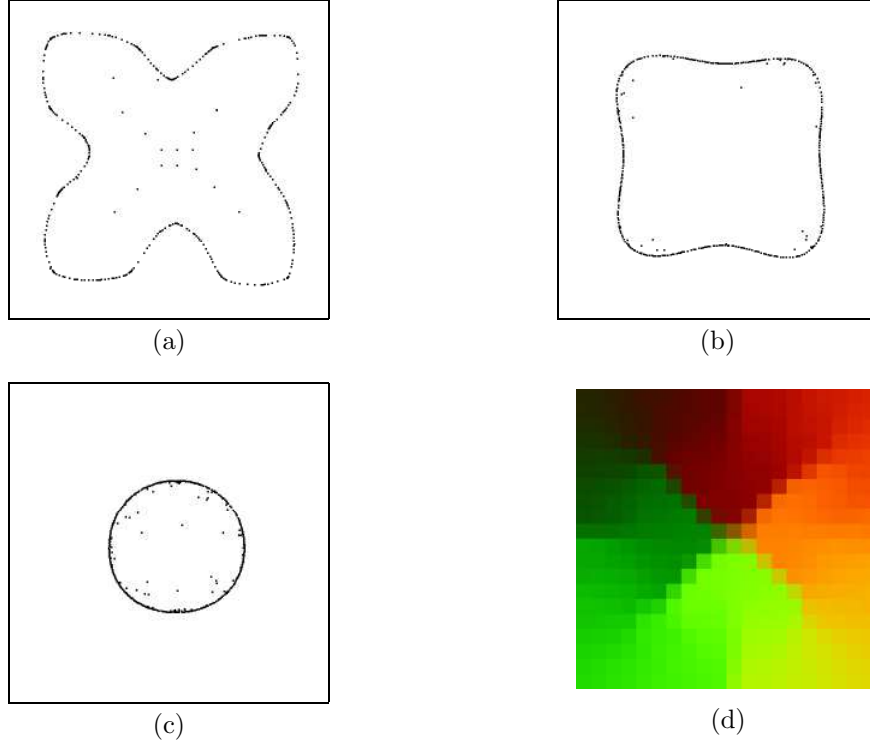


Figura 6.82: Proyecciones del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a una circunferencia y encadenado basado en un código de colores lineal. Parámetros utilizados:  $\theta = 0,25$  y  $r = 20$  en (a),  $\theta = 25$  y  $r = 1$  en (b), y  $\theta = 150$  y  $r = 1$  en (c). La imagen coloreada (d) corresponde al encadenado de la proyección del MAO con la de la imagen (c).

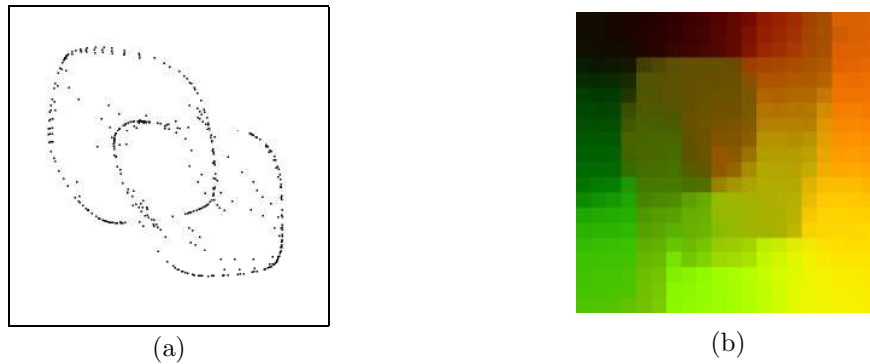


Figura 6.83: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados.

es generada con el núcleo de vecindad descrito en (6.20), donde el conjunto de parámetros corresponde al “estándar”, salvo el número de iteraciones que es incrementado hasta  $T = 500$ . A diferencia de los métodos clásicos la proyección consigue desvelar la dimensión intrínseca de los datos, y aporta una idea sobre la forma real de las clases (circular y continua). La imagen coloreada es difícil de interpretar ya que no aparecen *clusters* en los datos y el cruce de las dos circunferencias en el plano otorga el mismo color a neuronas pertenecientes a clases diferentes. En cualquier caso, es posible discriminar ambas clases con la ayuda de la proyección u otros métodos de visualización (U-matrix). Por otro lado, la continuidad suave de los colores ayuda a deducir la forma cíclica de los datos y una posible distribución uniforme de éstos. Por otro lado, la Figura 5.40 (a) muestra la proyección generada con el método de contracción de Himberg donde  $\theta = 1$  y  $r = 1$ , y el correspondiente encadenado mediante un código de colores lineal. Estas imágenes vuelven a ser muy similares a las generadas por el AAN.

#### 6.3.3.11. Círculo y anillo

La dimensión intrínseca de la distribución formada por un círculo y un anillo es claramente dos, por lo que un MAO bidimensional debería adaptarse adecuadamente a esta distribución. Sin embargo, resulta difícil entrenar un MAO de modo que preserve perfectamente la topología debido a la fuerza de atracción del círculo central, lo cual provoca que varias neuronas del perímetro exterior del MAO representen datos del círculo. De esta manera, el anillo queda representado en el exterior del MAO formando una figura con forma de “C”. Esto puede ser apreciado con claridad observando directamente los datos y las conexiones de los vectores de referencia en el espacio original de los datos (en este caso observable), véase la Figura 6.33, donde el MAO es  $(20 \times 20)$ -dimensional. Las imágenes generadas por la mayoría de métodos de visualización de MAO también revelan este defecto (véase la Figura 6.44).

Los métodos de contracción superan el problema anterior ya que las neuronas del exterior del MAO que representan al círculo son atraídas rápidamente hacia el centro de la proyección donde éste queda representado. Las proyecciones ilustran adecuadamente un anillo exterior continuo y un *cluster* central compacto, donde existe una mayor concentración en su centro, tal y como sucede en la distribución original de los datos. La Figura 6.84 muestra la proyección del AAN para el MAO  $(20 \times 20)$ -dimensional, generada con el núcleo de vecindad descrito en (6.20),

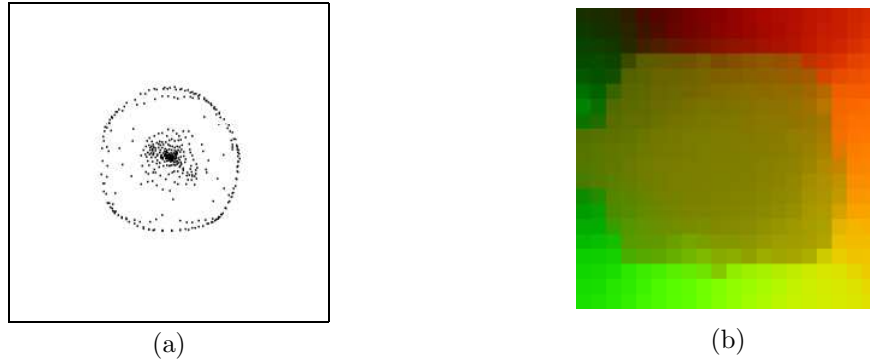


Figura 6.84: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un círculo y un anillo concéntricos.

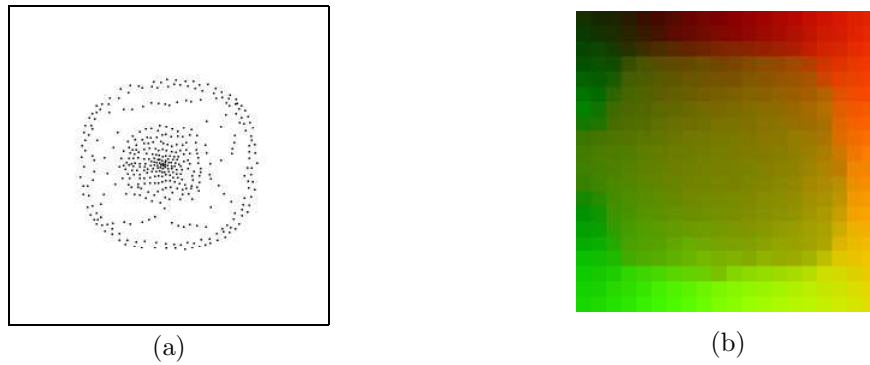


Figura 6.85: Proyección generada con el método de contracción de Himberg con  $\theta = 2$  y  $r = 1$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por un círculo y un anillo concéntricos.

donde el número de iteraciones es relativamente elevado  $T = 1000$  para obtener mayor precisión en la representación de la distribución (a). El resto de parámetros corresponden a los “estándar”. El encadenado con la proyección original del MAO es mostrado en (b). En esta imagen coloreada aparece un *cluster* central ilustrado mediante un color uniforme, que lo separa claramente de la clase exterior. La continuidad del anillo exterior, que no puede ser detectada mediante técnicas tradicionales de análisis de MAO, queda representada por la transición continua de los colores en el perímetro del MAO. Obsérvese que las neuronas adaptadas al anillo donde se produce la discontinuidad aparecen con un color similar en el mapa.

Por otro lado, la Figura 6.85 muestra la proyección generada con el método de contracción de Himberg (a), donde  $\theta = 2$  y  $r = 1$ . El correspondiente encadenado

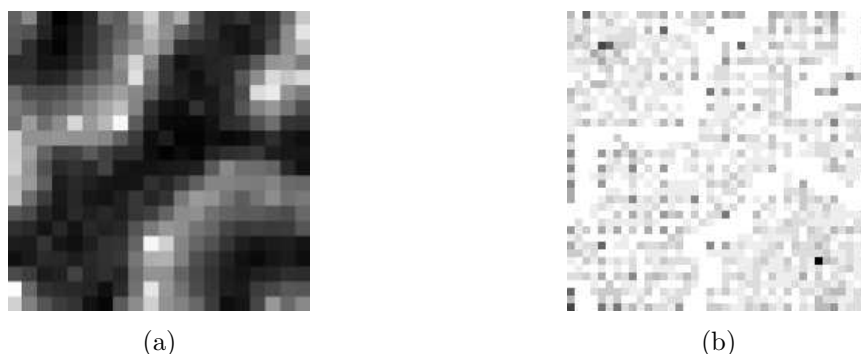


Figura 6.86: U-matrix (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca.

mediante un código de colores lineal es mostrado en (b). Las imágenes son muy similares a las obtenidas mediante el AAN (véase la Figura 6.84). Aumentando cualquiera de los dos parámetros del algoritmo es posible generar una proyección muy precisa de la distribución de datos. Sin embargo, en este caso, a medida que aumenta la precisión de la proyección resulta más complicado detectar el borde entre el círculo y el anillo en el encadenado basado en un código de colores. Nótese que resulta más fácil discriminar las dos clases en el encadenado generado a partir del AAN.

#### 6.3.3.12. Esfera maciza dentro de una esfera hueca

La distribución formada por una esfera maciza dentro de una esfera hueca es la “versión tridimensional” de la distribución formada por un círculo y un anillo. La dimensión intrínseca de la esfera maciza es claramente tres, por lo que un MAO bidimensional presenta serias dificultades para adaptarse ordenadamente a ella. El grosor de la esfera hueca es lo suficientemente delgado como para considerar que los datos se hallan aproximadamente sobre una superficie bidimensional, y el MAO podrá adaptarse mejor a ellos que a la esfera maciza. Sin embargo, la forma “cerrada” de la esfera hueca genera discontinuidades en el MAO bidimensional (especialmente en sus bordes exteriores dado que éste representa una superficie acotada), al igual que el desorden de las neuronas que representan a la esfera maciza. De esta manera, la información contenida en el MAO y visualizada mediante los métodos de análisis generalmente no es fiable y resulta muy difícil de interpretar.

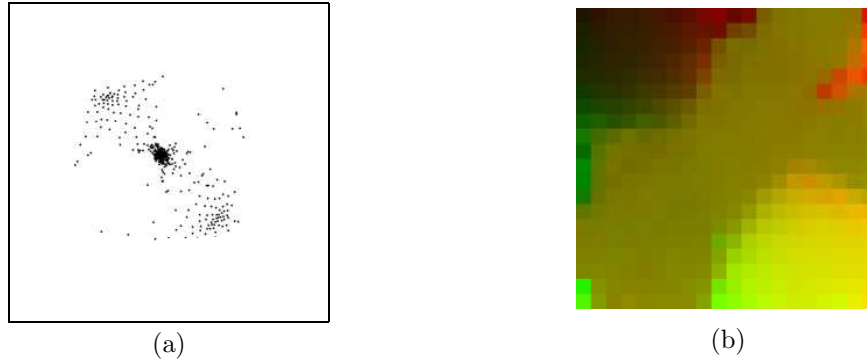


Figura 6.87: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca.

La Figura 6.86 muestra la U-matrix y el histograma de datos (creado a partir de la variante descrita en la Tabla 5.3) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca (véase la Figura 6.31), en (a) y (b), respectivamente. En este caso, la U-matrix y el histograma de datos ilustran tres zonas oscuras disjuntas, que pueden ser interpretadas como *clusters*. La clase central a lo largo de la diagonal secundaria ( $45^\circ$ ) representa al *cluster* de la esfera maciza, mientras las otras dos clases representan a la esfera hueca, que no son *clusters*. El analista tiene la difícil tarea de visualizar que los datos de estas dos clases realmente pertenecen a una sola, a pesar de estar ubicados en extremos opuestos del MAO y de la propia distribución. Este ejemplo demuestra la necesidad de comprobar la calidad de la adaptación de los MAO a los conjuntos de datos mediante medidas de bondad relacionadas con la preservación de la topología, para determinar si el MAO resulta adecuado para modelar y examinar visualmente los datos antes de ser utilizado. En este caso, la dimensión del MAO no permite analizar adecuadamente la distribución.

Sin embargo, los vectores de referencia pueden ser proyectados a un plano bidimensional de manera que se pierda la estructura regular de la proyección que genera el MAO. Una opción consiste en utilizar métodos de contracción, que además facilitan la interpretación de las imágenes producidas por los métodos tradicionales. La Figura 6.87 muestra la proyección del AAN generada con el núcleo de vecindad descrito en (6.18), donde el número de iteraciones es elevado a  $T = 5000$ , la tasa de aproximación es reducida a  $\lambda(0) = 0,005$  y  $\tau$  es incrementado a 0,55, para obtener mayor precisión en la representación de la distribución

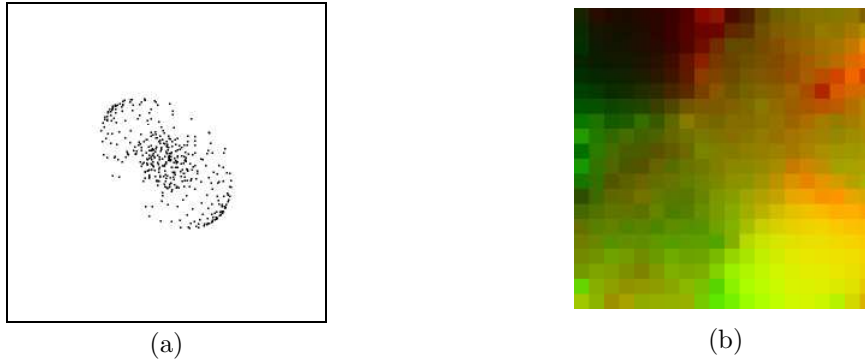


Figura 6.88: Proyección generada con el método de contracción de Himberg con  $\theta = 3$  y  $r = 1$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca.

(a). Al estar diseñado principalmente para analizar *clusters*, la proyección consigue agrupar los vectores de referencia asociados a la esfera maciza en un *cluster* compacto, mientras que el resto quedan repartidos por el espacio observable. Esta agrupación permite detectar claramente la clase que representa al *cluster* central en la imagen coloreada fruto del encadenado mediante un código de colores lineal de la proyección con la original del MAO (b). La esfera maciza queda representada mediante un color uniforme (verde-oliva), quedando claramente separada del resto de clases en el mapa. Por otro lado, a pesar de que las neuronas que representan a la esfera hueca aparecen en diversas regiones disjuntas en el MAO, es posible detectar una relación entre ellas a través de las similitudes en los colores de sus bordes. Este fenómeno queda aún más claro en la imagen coloreada asociada al método de Himberg, véase la Figura 6.88 (b).

La Figura 6.88 muestra la proyección generada con el método de contracción de Himberg (a), donde  $\theta = 3$  y  $r = 1$ , la cual es más precisa que la generada con el ANN. El correspondiente encadenado mediante un código de colores lineal es mostrado en (b). La imagen coloreada facilita la interpretación de las relaciones entre las diferentes regiones del MAO asociadas a la esfera hueca. Por ejemplo, el borde derecho del “*cluster*” superior-izquierdo recibe un color anaranjado, al igual que una pequeña región de la esquina superior-derecha y el borde superior del “*cluster*” inferior-derecho, indicando que todas estas zonas representan datos similares. Con los tonos verdes claros sucede un fenómeno similar en el extremo opuesto del mapa. Por otro lado, la no-uniformidad de los colores de las neuronas asociadas a la esfera maciza indica las numerosas discontinuidades del MAO al





Figura 6.89: Matriz de distancias (a) e histograma de datos (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos superficies cuadradas planas paralelas.

adaptarse a la distribución tridimensional.

#### 6.3.3.13. Dos superficies planas paralelas

En este caso la distribución a analizar está compuesta por dos superficies cuadradas planas paralelas. El experimento resulta muy similar al llevado a cabo con la distribución formada por dos segmentos. Un MAO bidimensional es capaz de discriminar las dos clases, pero no aporta información válida sobre la forma de los datos ni su dimensión intrínseca. La Figura 6.89 ilustra la matriz de distancias y el histograma de datos para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución (véase la Figura 6.26) en (a) y (b), respectivamente. Nótese la similitud entre estas imágenes y las ilustradas en la Figura 6.76. Las superficies cuadradas quedan claramente separadas por un borde claro, mientras la poca variación de los tonos de gris indica una posible distribución uniforme de los datos en ambas clases.

La Figura 6.90 muestra la proyección del AAN, para el mismo MAO  $(20 \times 20)$ -dimensional, generada con el núcleo de vecindad descrito en (6.20) (a), donde el número de iteraciones es elevado a  $T = 500$  y el resto de parámetros corresponden a los “estándar”. En la proyección es posible intuir que las dos clases o *clusters* representan superficies donde los datos aparecen según una distribución uniforme. El correspondiente encadenado produce la imagen coloreada ilustrada en (b), la cual es muy similar a las imágenes generadas con los métodos tradicionales de visualización (véase la Figura 6.89) y consigue discriminar adecuadamente ambas clases. Por otro lado, la transición suave y constante de los colores también indica la uniformidad de los datos en cada clase.

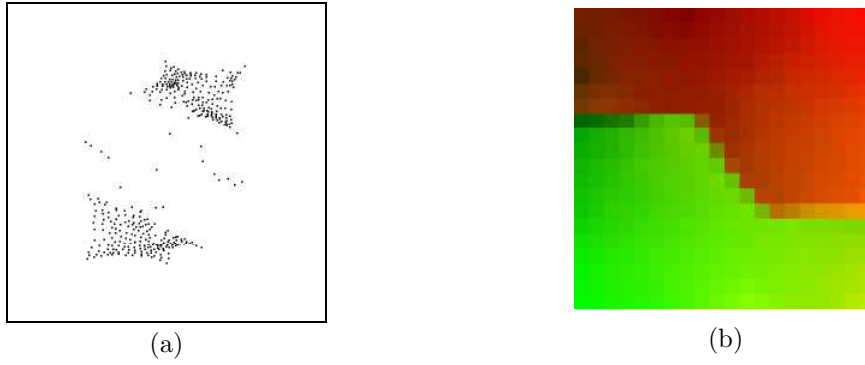


Figura 6.90: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos superficies cuadradas planas paralelas.

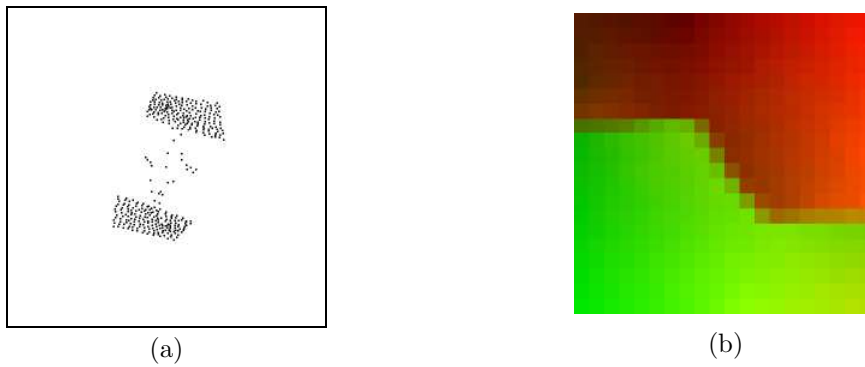


Figura 6.91: Proyección generada con el método de contracción de Himberg con  $\theta = 0,5$  y  $r = 1$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos superficies cuadradas planas paralelas.



Figura 6.92: U-matrix (a) e histograma de datos (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución con forma de estrella.

La Figura 6.91 muestra la proyección generada con el método de contracción de Himberg (a), donde  $\theta = 0,5$  y  $r = 1$ . El correspondiente encadenado mediante un código de colores lineal es mostrado en (b). Nótese la similitud entre estas imágenes y las ilustradas en la Figura 6.90, donde la uniformidad de las clases queda ligeramente más clara en la proyección generada por el método de Himberg.

#### 6.3.3.14. Estrella

La siguiente distribución tiene forma de estrella y está compuesta por tres ortoedros delgados, dispuestos según las direcciones de los ejes de coordenadas en  $\mathbb{R}^3$ . La dimensión intrínseca de los datos es tres, por lo que aparecen numerosas discontinuidades locales en MAO bidimensionales. No obstante, éstos son capaces de desvelar la forma global de los datos (sus seis “brazos”). La Figura 6.92 ilustra la U-matrix y el histograma de datos (generado a partir de la variante descrita en la Tabla 5.3) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución (véase la Figura 6.34) en (a) y (b), respectivamente. Una vez más, la poca variación de los tonos de gris indica que la distribución de los datos es uniforme.

La Figura 6.93 muestra la proyección del AAN para el mismo MAO  $(20 \times 20)$ -dimensional (a), y el correspondiente encadenado (b). La proyección es generada con el núcleo de vecindad descrito en (6.20), donde el conjunto de parámetros es el “estándar”. Ésta ilustra adecuadamente las estructuras que componen la distribución en dos dimensiones (mejorando la visualización aportada por los métodos tradicionales), donde es posible distinguir el ligero grosor de los seis “brazos” de la estrella. La imagen coloreada fruto del encadenado es similar a la aportada por los métodos tradicionales (véase la Figura 6.92), y muestra cada

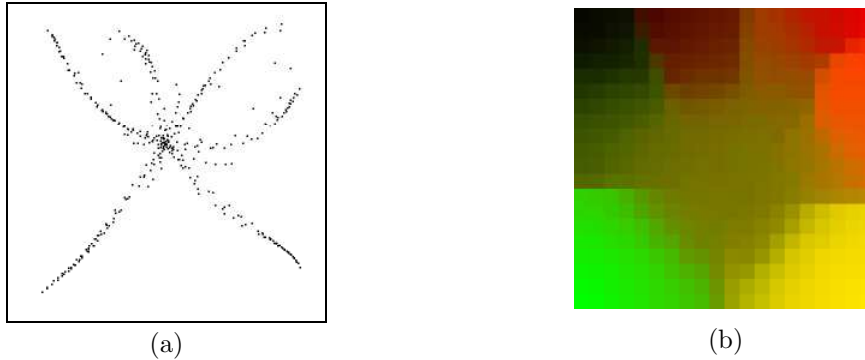


Figura 6.93: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución con forma de estrella.

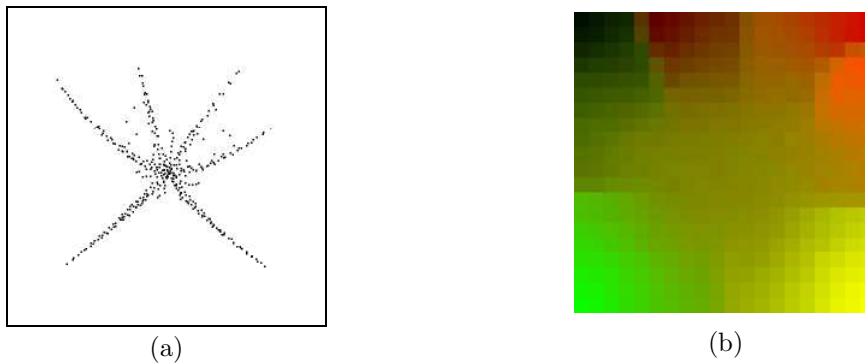


Figura 6.94: Proyección generada con el método de contracción de Himberg con  $\theta = 0,5$  y  $r = 1$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución con forma de estrella.

segmento o brazo con un color diferente, indicando la diferencia entre éstos. Sin embargo, la imagen resulta difícil de interpretar correctamente ya que a simple vista aparecen siete zonas con colores diferentes. La continuidad entre los brazos de la estrella y su centro no es evidente, por lo que el analista puede pensar que existe un *cluster* central, separado de las clases situadas alrededor de éste. Por esta razón el analista debe examinar al mismo tiempo la propia proyección, que es la que aporta la información necesaria sobre la continuidad de los datos.

El problema anterior también puede ser mitigado aumentando ligeramente la precisión de la proyección, lo cual puede ser conseguido aumentando las iteraciones del AAN o utilizando el método de Himberg. La Figura 6.94 muestra la proyección generada con el método de contracción de Himberg (a), donde  $\theta = 0,5$  y  $r = 1$ . El correspondiente encadenado mediante un código de colores lineal es

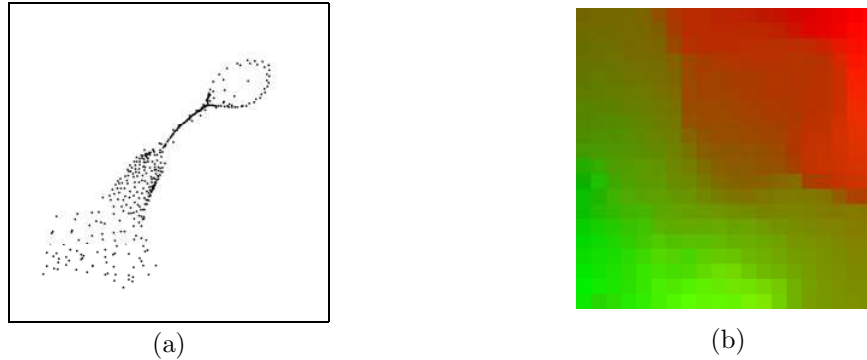


Figura 6.95: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro.

mostrado en (b). Nótese la similitud entre estas imágenes y las ilustradas en la Figura 6.93. En este caso, la continuidad de los datos puede ser apreciada con claridad en la imagen coloreada.

### 6.3.3.15. Circunferencia, segmento, cuadrado y ortoedro

Es posible ver en el experimento llevado a cabo en la Sección 6.2.3.6 que los métodos tradicionales de visualización tienen problemas para representar adecuadamente las características de la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro. La superposición de la U-matrix con el histograma de datos ayuda a interpretar correctamente las diversas estructuras de la distribución, véase la Figura 6.48. No obstante, los métodos de proyección consiguen generar una descripción más clara de los datos, como el MST (véase la Figura 6.35) o los métodos de contracción, especialmente cuando su dimensión intrínseca es menor que la dimensión del MAO.

La Figura 6.95 muestra la proyección del AAN para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución (a), véase la Figura 6.35. El correspondiente encadenado es ilustrado en (b). La proyección es generada con el núcleo de vecindad descrito en (6.20), donde el conjunto de parámetros corresponde al “estándar”. La proyección ilustra todas las clases de los datos con suficiente precisión, teniendo en cuenta que es imposible representar adecuadamente el ortoedro en dos dimensiones. Por otro lado, aunque la proyección puede ayudar a interpretar la imagen coloreada fruto del encadenado, en este caso resulta muy difícil de interpretar, a pesar de que cada color en el mapa está asociado a una clase

de datos (el rojo brillante a la circunferencia, el rojo oscuro al segmento, el verde oscuro al cuadrado y el verde brillante al ortoedro). Por ejemplo, no es posible detectar que las esquinas superior-izquierda e inferior-derecha apenas representan datos. Tampoco es posible apreciar con claridad una forma circular en la esquina superior-derecha, ni una estructura lineal para el segmento.

Por otro lado, la Figura 5.40 (b) muestra la proyección generada con el método de contracción de Himberg donde  $\theta = 3$  y  $r = 1$ , y el correspondiente encadenado mediante un código de colores lineal. Estas imágenes vuelven a ser muy similares a las generadas por el AAN.

#### 6.3.3.16. Datos reales de habla

La complejidad de los datos reales de habla es considerablemente mayor que la relativa a las distribuciones de simulación. Suelen presentar casos atípicos, ruido y un solapamiento de clases notable debido a su alta variabilidad, lo cual dificulta su análisis. Al estar definidos en  $\mathbb{R}^8$ , sólo es posible visualizarlos mediante técnicas de proyección a un espacio observable, lo cual implica una pérdida de información. Aunque la dimensión intrínseca de los datos reales sea menor o igual a tres (la de los datos de habla utilizados en [BHV99] está comprendida entre dos y tres según un estudio con medidas de bondad) generalmente no es posible conocer sus características *a priori*, por lo que no resultan adecuados para evaluar técnicas de visualización. Por otro lado, el conocimiento de un experto sobre datos de habla y sobre la propia herramienta de visualización es necesario para realizar un análisis efectivo y poder llegar a una conclusión en cuanto a la utilidad de dicha herramienta.

La Sección 6.2.3.7 presenta un experimento donde los datos reales de habla son analizados mediante la superposición de una U-matrix y un histograma de datos (véase la Figura 6.49). Estas imágenes aportan información sobre la estructura de los *clusters* de la distribución en el mapa. Sin embargo, la mayor fuente de información relativa a la localización de los diferentes fonemas o “subfonemas”, que permite etiquetar el MAO, analizar secuencias de fonemas, detectar las clases existentes y su relación, e incluso detectar discontinuidades en el MAO, es aportada por las imágenes coloreadas que generan los métodos de contracción de neuronas mediante el encadenado.

La Figura 6.96 muestra la proyección del AAN para un MAO  $(16 \times 16)$ -dimensional adaptado a la distribución formada por datos reales de habla de la

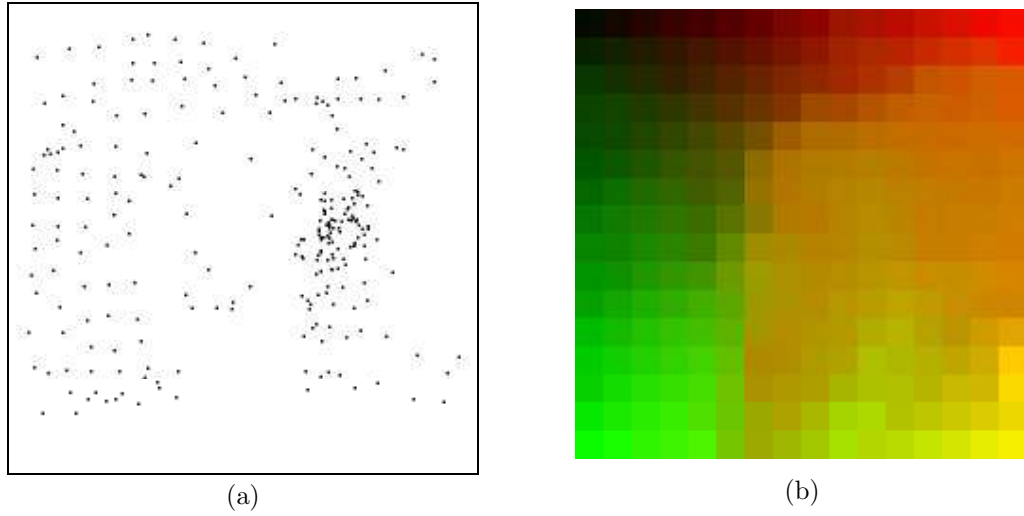


Figura 6.96: Proyección generada con el AAN (a) y correspondiente encadenado mediante un código de colores lineal (b), para un MAO ( $16 \times 16$ )-dimensional adaptado a la distribución formada por datos reales de habla de la palabra “right”.

palabra “right” (a), véase la Figura 6.49. El correspondiente encadenado es ilustrado en (b). La proyección es generada con el núcleo de vecindad descrito en (6.20), donde el conjunto de parámetros corresponde al “estándar”, excepto el número de iteraciones que es elevado a  $T = 500$ . La proyección atrae principalmente a las neuronas asociadas a los dos *clusters* densos claramente visibles en la U-matrix (que corresponden a dos clases de silencio). Esto indica que dichas clases representan datos muy similares. Por otro lado, en la parte izquierda del mapa la atracción entre las neuronas es mucho menor, lo cual indica una mayor dispersión de datos en las clases representadas por dichas neuronas (vocales y el fonema /r/). Esto naturalmente queda reflejado en la imagen coloreada generada a partir del encadenado. El silencio queda representado por una clase de color naranja (separada del resto), mientras no aparece otra zona de color constante en el resto del mapa.

Dado el alto grado de dispersión y solapamiento de las clases en los datos de habla (no existen *clusters* fáciles de discriminar excepto el silencio), la verdadera utilidad de la proyección consiste en aplicar el encadenado y obtener una imagen coloreada donde cada clase quede representada por una familia de colores. Posteriormente, a cada patrón de la distribución es posible asignarle el color asociado a su correspondiente neurona ganadora. De esta manera, surgen nuevas visualizaciones de los datos que ayudan a llevar a cabo el proceso de análisis.

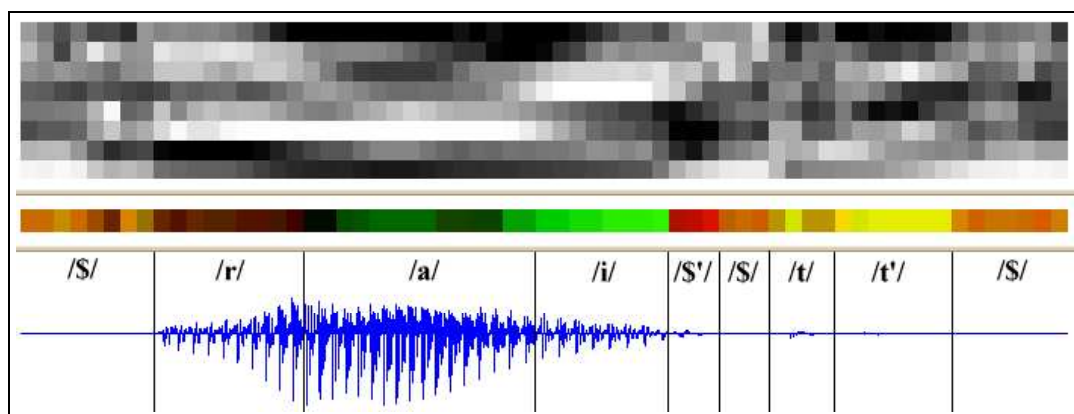


Figura 6.97: Análisis de secuencias mediante símbolos coloreados. Cada vector de ocho parámetros *RASTA-PLP* (los valores de sus componentes son mostrados por columnas mediante niveles de gris en orden ascendente) correspondiente a 10 milisegundos de una traza de la palabra “right”. Cada vector puede ser representado (en este caso mediante rectángulos) en una lista coloreada, donde recibe el color asociado a su neurona ganadora en el mapa coloreado.

Dada la naturaleza dinámica de los datos de habla, una de las aplicaciones principales del mapa coloreado consiste en analizar secuencias de datos, lo cual va a permitir etiquetar tanto la señal de habla como el propio MAO. La Figura 6.97 muestra la misma señal de la palabra “right” y secuencia de vectores de parámetros *RASTA-PLP* asociados, previamente ilustradas en la Figura 4.16. La figura también muestra una lista de rectángulos coloreados asociados a cada patrón de entrada, donde su color corresponde al de la neurona ganadora de dicho patrón en el mapa coloreado. Mediante esta lista es posible distinguir los diferentes fonemas y “subfonemas” que aparecen en la señal de habla de forma visual, ya que varias secciones de la lista aparecen con un color similar. Un análisis detallado de la señal indica que el silencio /s/ queda representado por colores anaranjados y castaños claros. El negro y marrón oscuro corresponden al fonema /r/. Los verdes al diptongo formado por los fonemas /a/ (verde oscuro) e /i/ (verde claro). El rojo oscuro corresponde a una señal de poca energía /s'/, parecida al silencio, que aparece generalmente al final del fonema /i/ (el rojo brillante de la esquina superior-derecha del mapa coloreado corresponde a otro tipo diferente de silencio, poco frecuente, que resulta ser una clase atípica, ya que un porcentaje elevado de los datos asociados a ese color aparecen en una sola traza). Por último, es posible dividir la “t” en dos clases o “subfonemas” diferentes /t/ y /t'/, asociados a colores marrones claros y amarillos, respectivamente.

La imagen coloreada no sólo sirve para detectar los *clusters* o clases y sus



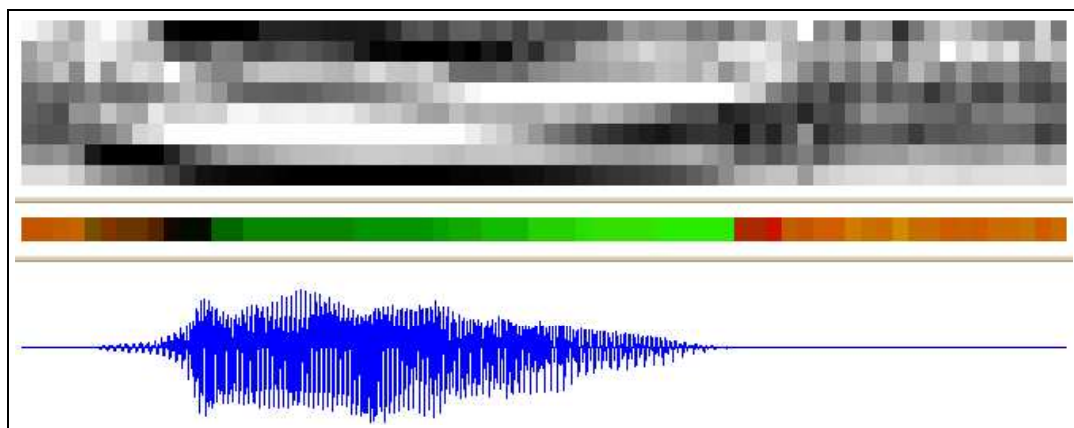
bordes. El ejemplo también muestra el poder exploratorio de los MAO, ya que permite detectar ciertas características inesperadas en los datos, como eran el silencio asociado al rojo brillante, el final del fonema /i/, los dos grandes *clusters* separados de silencio, o la división del fonema /t/.

La lista o secuencia coloreada de la Figura 6.97, relativa a un locutor masculino grabado sin ruido inducido, es similar a la mayoría de secuencias obtenidas con el resto de trazas de la base de datos de la palabra “right”. De esta manera, el uso del mapa coloreado ayuda a etiquetar toda la base de datos (el etiquetado de una base de datos de habla es un problema tremendamente complejo incluso para un especialista), y el proceso puede ser realizado a la inversa para etiquetar las neuronas del MAO.

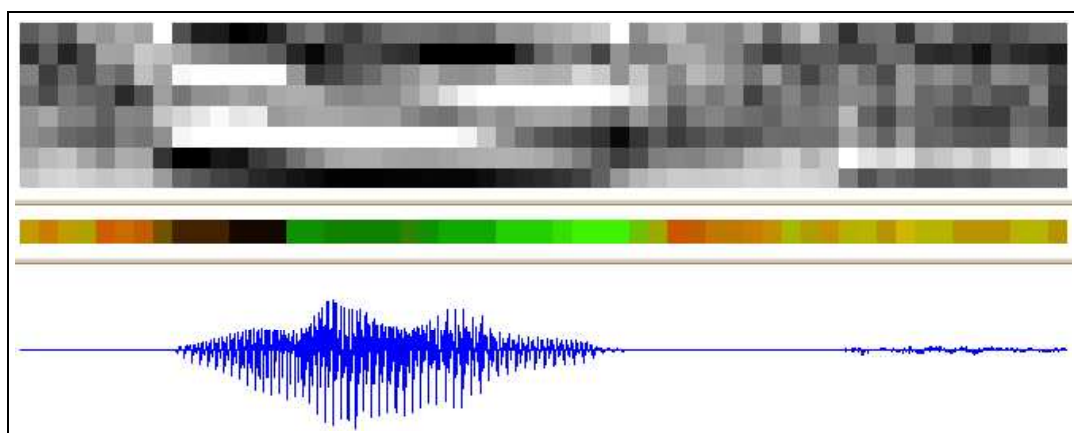
La Figura 6.98 muestra tres ejemplos de las secuencias coloreadas obtenidas a partir de trazas de la base de datos de habla. Aunque la mayoría de trazas corresponden a locutores españoles (no nativos), en el primer ejemplo el locutor, que es grabado con ruido inducido, tiene un acento claramente estadounidense (en concreto de la ciudad de Nueva York, que es uno de los ocho principales acentos estadounidenses). La diferencia con respecto al resto de trazas de la base de datos resulta en que apenas pronuncia el fonema /t/, por lo que no aparece una secuencia amarilla al final de la secuencia, aunque es posible apreciar algunos elementos marrones claros (a). En (b) la traza corresponde a una mujer grabada sin ruido inducido, y al contrario del primer ejemplo pronuncia la “t” de forma exagerada, por lo que al final de la secuencia aparece una zona extensa amarilla y marrón clara. El último ejemplo incluye una traza donde aparecen numerosos datos atípicos, y es la responsable de la mayor parte de datos asociados a la esquina superior-derecha del mapa, coloreada con un rojo brillante. Al principio de la traza también aparecen datos atípicos antes del fonema /r/, donde el sonido parece el principio de una “g”.

Por otro lado, resulta más sencillo etiquetar la base de datos y el MAO mediante este tipo de visualización que por medio de grafos (véase la Sección 5.2.4). La Figura 6.99 (a) muestra un grafo donde aparecen conectadas las neuronas ganadoras de los vectores de datos adyacentes ilustrados en la Figura 6.97. El mismo grafo puede ser mostrado aplicando el mapa coloreado como fondo (b), lo cual facilita su análisis.

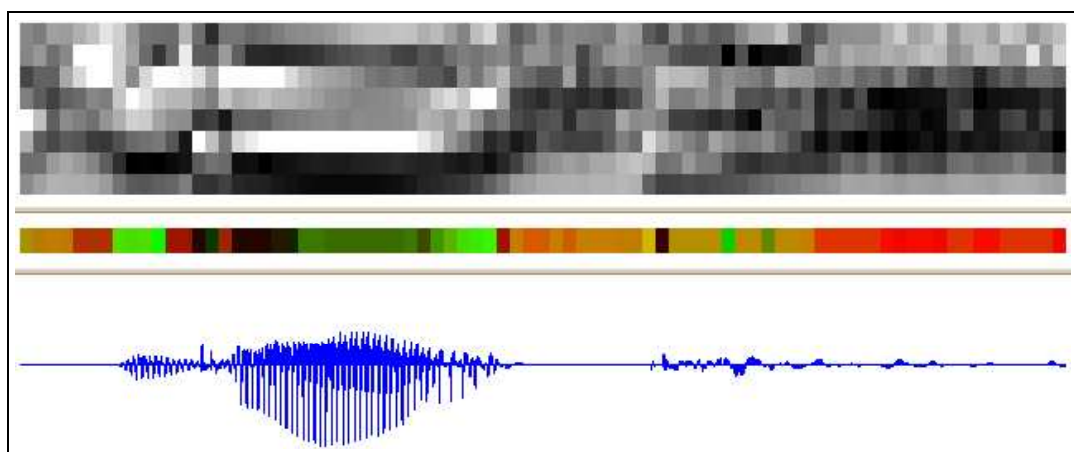
Con respecto al método de Himberg, éste sigue generando proyecciones similares a las obtenidas por el AAN. La Figura 6.100 muestra la proyección generada



(a)



(b)



(c)

Figura 6.98: Ejemplos de secuencias coloreadas. La traza en (a) corresponde a un locutor masculino grabado con ruido inducido, en (b) a una mujer sin ruido inducido y en (c) a una mujer con ruido inducido.

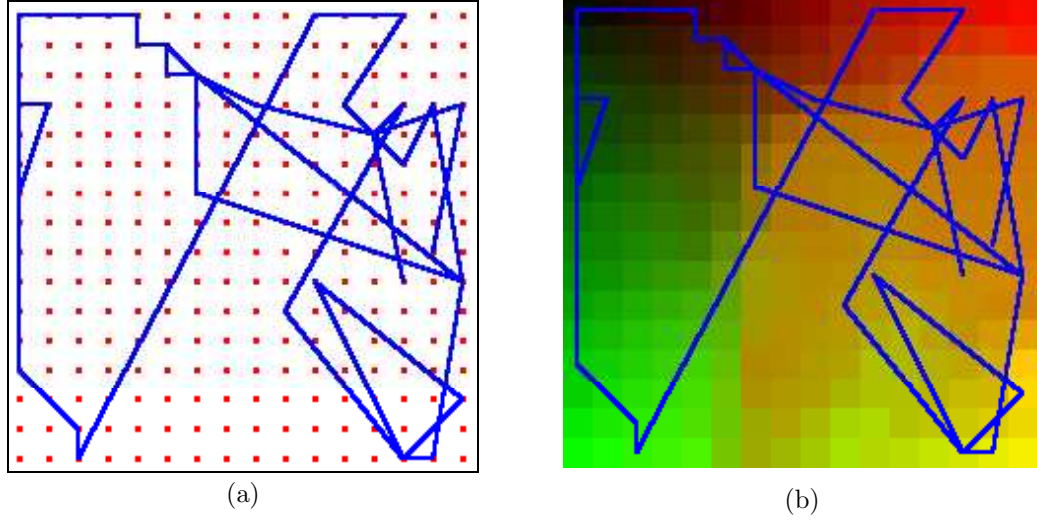


Figura 6.99: Análisis de la secuencia de datos de la Figura 6.97 mediante un grafo que conecta neuronas ganadoras de datos adyacentes (a). El mapa coloreado generado a partir del encadenado puede ser utilizado como fondo para facilitar el análisis (b).

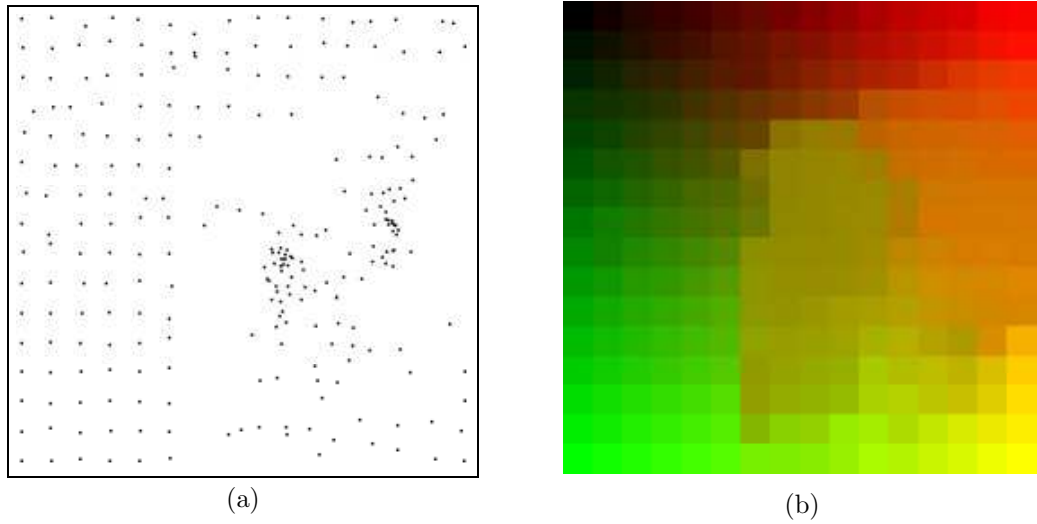


Figura 6.100: Proyección generada con el método de contracción de Himberg con  $\theta = 0,25$  y  $r = 25$  (a), y correspondiente encadenado mediante un código de colores lineal (b), para el MAO  $(16 \times 16)$ -dimensional adaptado a la distribución formada por datos reales de habla de la palabra “right”.

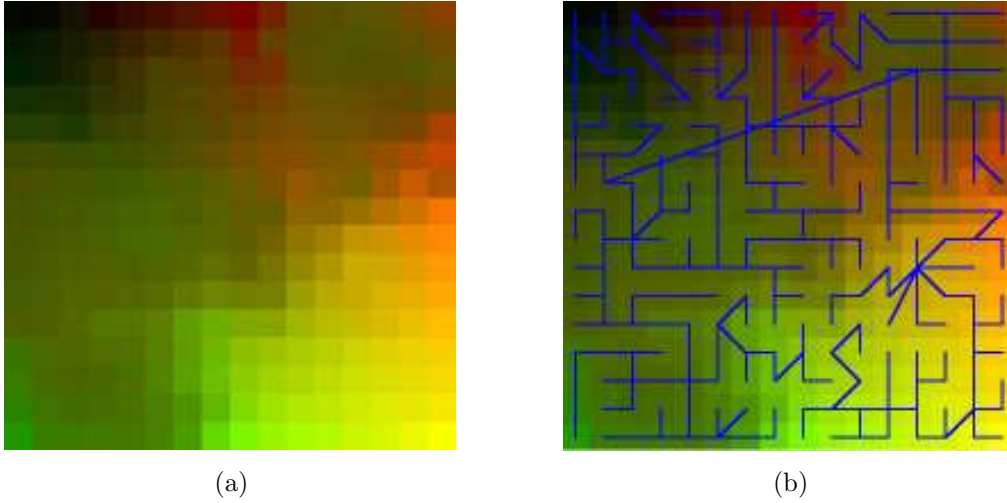


Figura 6.101: Uso de mapas coloreados como medidas de bondad. La imagen (a) corresponde a un mapa coloreado (tras aplicar el AAN y el encadenado a una base de datos de habla) donde aparecen tres *clusters* de color verde que representan silencio. Sin embargo, los dos *clusters* superiores están adaptados a los mismos datos, como muestra la arista larga de la proyección del árbol generador mínimo (que conecta vectores de referencia en el espacio de los datos) sobre el mapa (b).

con el método de contracción de Himberg (a), donde  $\theta = 0,25$  y  $r = 25$ . El correspondiente encadenado mediante un código de colores lineal es mostrado en (b). Los parámetros han sido escogidos para que el método represente los dos grandes *clusters* de silencio mediante colores diferentes.

La Sección 6.3.3.12 muestra un ejemplo donde es posible detectar discontinuidades en un MAO por medio de las asociaciones de colores relativas a los métodos de contracción. Éstas aparecen cuando neuronas lejanas en el mapa presentan colores similares. La Figura 6.101 muestra un ejemplo de un mapa coloreado (tras aplicar el AAN y el encadenado) donde es posible observar tres clases de silencio cuando en realidad sólo existen dos (a). El ejemplo está basado en un MAO  $(16 \times 16)$ -dimensional adaptado a una distribución de habla de la palabra inglesa “up”, grabada en las mismas condiciones y por los mismos locutores que la de la palabra “right”. En las imágenes es posible detectar tres *clusters* de color verde-oliva (correspondientes a datos de silencio). Sin embargo, un análisis más profundo indica que los dos *clusters* superiores representan prácticamente los mismos datos. Es posible ver esta discontinuidad del MAO tras proyectar el árbol generador mínimo, que conecta a vectores de referencia en el espacio de los datos sobre el propio mapa. Una de las aristas es considerablemente más larga

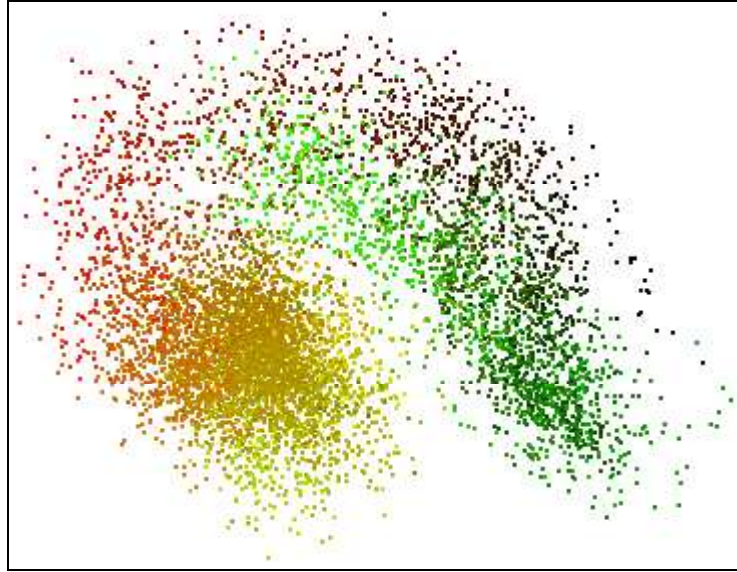


Figura 6.102: Proyección coloreada de los datos de habla según sus dos primeras componentes principales.

que las demás, indicando un par de vectores de referencia próximos en el espacio de los datos pero lejos en el mapa. También es posible obtener MAO con *clusters* de silencio “repetidos” con la base de datos de la palabra “right”.

### 6.3.3.17. Combinación del AAN con el MST

La asignación de colores a los datos también puede ser aprovechada para colorear proyecciones de éstos a espacios observables. La Figura 6.102 muestra la proyección coloreada de la distribución de la palabra “right” según sus dos primeras componentes principales (véase la proyección sin colorear en la Figura 6.49 (d)). El color aporta profundidad a la imagen, donde es posible observar la misma transición de colores que aparece en el mapa coloreado (verde claro, verde oscuro, marrón oscuro, rojo oscuro, rojo, naranja, marrón claro y amarillo).

El MST crea proyecciones no-lineales de los datos que pueden ser coloreadas siguiendo la misma estrategia. La Figura 6.103 muestra dos proyecciones coloreadas del método para la distribución de habla. La proyección ilustrada en (a) está basada en el error entre triángulos  $\xi_1$ , mientras la mostrada en (b) lo está en  $\xi_2$ . En ambos casos  $\Psi(x)$  contiene todos los pares de neuronas posibles y  $\varphi$  es escogida según (6.6). La proyección en (b) se ajusta más a la estructura del MAO

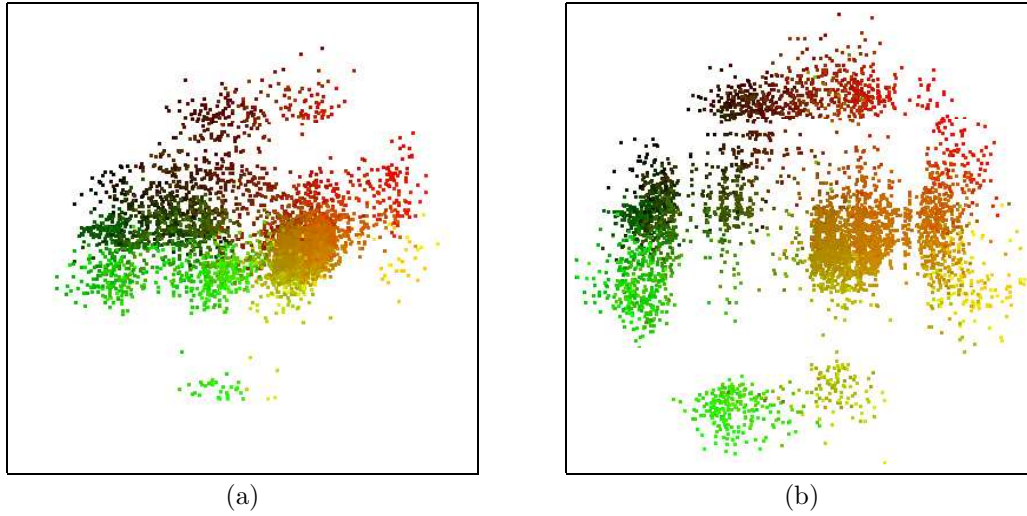


Figura 6.103: Proyecciones coloreadas de los datos de habla mediante el MST. En (a) el método utiliza  $\xi_1$  y en (b)  $\xi_2$ . En ambos casos  $\Psi(x)$  contiene todos los pares de neuronas posibles y  $\varphi$  es escogida según (6.6).

ya que la propiedad de ajuste se satisface mejor con  $\xi_2$  que con  $\xi_1$ . Las proyecciones permiten discriminar las diferentes clases por medio de los colores asociados a cada dato y dan una pista sobre datos atípicos o clases inesperadas. Por ejemplo, llama la atención la separación de los datos verdes claros en las zonas inferiores de las imágenes. En (a) los datos asociados al silencio y al fonema /t/ son proyectados sobre la misma zona del espacio de observación, lo cual no es extraño ya que ambos tipos de datos corresponden a señales similares de poca energía. Esto también ocurre en la proyección lineal basada en los dos primeros componentes principales ilustrada en la Figura 6.102.

La Figura 6.104 muestra otro ejemplo de la proyección del MST para los datos de habla. En esta ocasión el método utiliza  $\xi_1$ ,  $\Psi(x)$  contiene todos los pares de neuronas formados con las vecinas de la ganadora, y  $\varphi$  es escogida según (6.6). De esta manera, las proyecciones son rápidas y satisfacen mejor la propiedad de ajuste que las mostradas en la Figura 6.103. En (a) la proyección sólo contiene datos relativos a trazas grabadas por locutores masculinos y en (b) por femeninos. Las diferencias entre ambas proyecciones son notables. El *cluster* de silencio situado más a la izquierda está presente sobre todo en las trazas grabadas por mujeres, mientras los datos claramente amarillos sólo aparecen en el mapa “masculino”. También es posible observar que los datos vocálicos de los hombres están situados más hacia el exterior del mapa o espacio de observación. Por otro lado, los datos



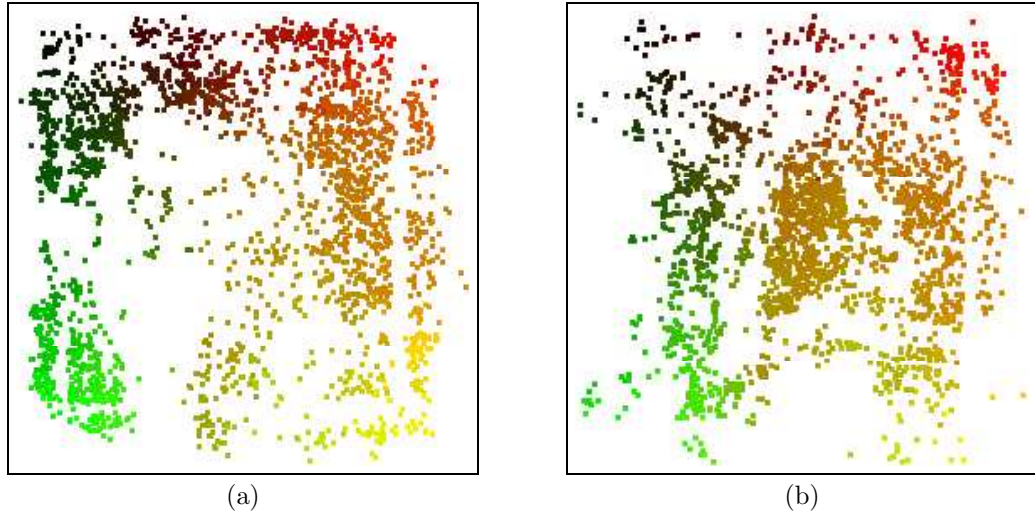


Figura 6.104: Diferencias entre hombres y mujeres mediante el MST. En (a) el método proyecta datos pertenecientes a trazas grabadas por hombres y en (b) por mujeres. En ambos casos el método utiliza  $\xi_1$ ,  $\Psi(x)$  contiene todos los pares de neuronas formados con las vecinas de la ganadora, y  $\varphi$  es escogida según (6.6).

atípicos del final de la traza ilustrada en la Figura 6.98 (c) están ubicados en la esquina superior-derecha del mapa “femenino”.

#### 6.3.4. Resultados experimentales: velocidad

Los experimentos anteriores confirman que el AAN puede producir resultados similares a los del método de contracción de Himberg, aunque las proyecciones generadas con este último son ligeramente más precisas. La ventaja del AAN sobre el método de Himberg radica en su velocidad cuando el MAO es de tamaño elevado o cuando el analista desea visualizar el proceso dinámico de contracción (el árbol jerárquico de clasificación).

Este apartado presenta un análisis de tiempos de ejecución relativos al AAN y al método de contracción de Himberg. El estudio ha sido llevado a cabo con MAO cuadrados de varios tamaños  $((20 \times 20), (32 \times 32), (64 \times 64), (100 \times 100))$ , adaptados a las distribuciones formadas por dos eslabones encadenados, ocho *clusters* disjuntos con forma hipercúbica, y cuatro *clusters* de igual tamaño y diferente forma. Los tiempos de ejecución presentados son relativos a un ordenador personal con procesador Intel® Pentium® 4 a 1.7GHz con 256MB de memoria RAM. El tiempo de pintado de las neuronas en las sucesivas iteraciones de los algoritmos no ha sido tenido en cuenta ya que no es relevante a la hora de

comparar los métodos. Sólo influye a la hora de visualizar el proceso dinámico de contracción, es muy dependiente del sistema, y el analista puede seleccionar la cantidad de proyecciones a visualizar independientemente del método de contracción utilizado. De esta manera, los tiempos sólo son relativos al cálculo de las sucesivas posiciones de las neuronas en cada iteración o proyección.

El análisis de la velocidad de un algoritmo no suele ser una tarea fácil. Existen multitud de factores tanto a nivel hardware como software a considerar a la hora de medir la velocidad de cualquier algoritmo (arquitectura del sistema, tipo de procesador, memoria RAM, memoria caché, compilador o intérprete, estructuras de datos, etc.). Por otro lado, la comparación entre el AAN y el método de contracción de Himberg es aún más compleja ya que éstos no generan exactamente los mismos resultados. Cada algoritmo está basado en una heurística particular, donde no existe una relación clara entre los parámetros utilizados (por ejemplo, la fuerza de atracción o el número de iteraciones), lo cual genera proyecciones de diferente precisión, a distinta velocidad. Por tanto, es necesario buscar una proyección de referencia para poder comparar ambos algoritmos.

El objetivo del AAN (cuya complejidad computacional está cerca de  $\mathcal{O}(M^{3/2})$ ) es crear proyecciones que describan las distribuciones lo más fielmente posible en el menor tiempo posible, aunque éstas no sean tan precisas como las generadas por otros métodos de contracción más lentos. De esta manera, con respecto a la proyección final del proceso, la referencia escogida en este estudio es la proyección que genera el método de Himberg para una sola iteración (donde su complejidad computacional es  $\mathcal{O}(M^2)$ ), que es la más rápida que puede generar el método y también la más precisa, como es posible ver en los experimentos llevados a cabo en la Sección 5.5.2.3. En cuanto al AAN, aunque es posible generar proyecciones que revelen adecuadamente características de las distribuciones con muy pocas iteraciones, la precisión aumenta a medida que lo hace el número de iteraciones. En general, 1000 iteraciones suelen ser suficientes para generar proyecciones precisas. Por otro lado, si la visualización del proceso dinámico de contracción forma parte del objetivo del análisis, el método de Himberg debe ser ejecutado en varias iteraciones (donde su complejidad computacional es  $\mathcal{O}(RM^2)$ ). En este caso, la referencia es el proceso de contracción del AAN para 1000 iteraciones, el cual garantiza una proyección final bastante precisa.

Las estructuras de datos y el tamaño de la memoria RAM son unos de los factores principales que intervienen en la velocidad de los algoritmos. Como éstos



calculan distancias entre vectores de referencia continuamente, dichas distancias pueden ser almacenadas en memoria, con lo que la velocidad del método aumenta, especialmente la del método de Himberg. El cálculo previo de las distancias entre los vectores de referencia puede representar un porcentaje elevado del tiempo total de ejecución, por lo que es previsible que la varianza de los tiempos de ejecución sea menor al implementar esta optimización. No obstante, la estructura que almacena las distancias debe caber en la memoria RAM, ya que el acceso a la memoria virtual (disco) puede ralentizar considerablemente el proceso. De esta manera, dos versiones diferentes de los algoritmos han sido probadas: una que almacena las distancias entre los vectores de referencia en memoria en una matriz simétrica, y otra que no las almacena y tiene que calcularlas repetidamente en cada iteración. La segunda versión de los algoritmos suele ser más lenta, salvo en algunos casos donde el número de iteraciones de los algoritmos es bajo, o especialmente si el tamaño del MAO es muy elevado y la estructura de distancias no cabe en la memoria RAM. En los experimentos el problema con la memoria RAM sólo afecta a los MAO  $(100 \times 100)$ -dimensionales, donde es necesario almacenar  $100^4 \cdot 4 \text{ bytes} \approx 381\text{MB}$  en memoria, que supera la disponible (256MB) en el sistema. Nótese que la información relativa a las distancias puede ser almacenada en una estructura no simétrica, donde no aparezca información redundante, lo cual reduciría a la mitad la cantidad de información a almacenar y el problema quedaría solucionado. No obstante, el uso de la matriz simétrica permite analizar esta situación, que puede surgir al trabajar con MAO todavía más grandes.

Cada experimento llevado a cabo analiza la velocidad del método de Himberg y del AAN al emplear los dos tipos de núcleos de vecindad propuestos. Cada proyección es calculada dos veces para medir los tiempos de ejecución al almacenar, o no, las distancias entre vectores de referencia en memoria. Para cada una de estas versiones de los algoritmos se calculan cinco proyecciones correspondientes a  $T = 200, 500, 1000, 2000$  y  $5000$  para el AAN; y  $r = 1, 10, 25, 50$  y  $100$  para el método de Himberg. La tasa de aproximación inicial  $\lambda(0)$  y el parámetro  $\theta$  son escogidos de tal forma que dichas cinco proyecciones resulten similares. Estos valores no necesitan ser modificados en función del tamaño del MAO. El resto de parámetros del AAN corresponden a los “estándar”. Los siguientes apartados ilustran tablas con los tiempos de ejecución obtenidos, además de una serie de gráficas para facilitar la comparación entre los dos métodos de contracción de neuronas.

6. NUEVOS MÉTODOS DE VISUALIZACIÓN  
DE MAPAS AUTO-ORGANIZATIVOS

	T	$\lambda(0)$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
AAN (6.18) Mem.	200	0,2	0,05	0,44	28,50	184,00
	500	0,1	0,11	0,49	28,67	170,32
	1000	0,05	0,16	0,61	28,89	173,45
	2000	0,03	0,22	0,77	29,44	179,11
	5000	0,01	0,55	1,21	30,98	215,69
AAN (6.18) No Mem.	200	0,2	0,05	0,55	30,04	43,01
	500	0,1	0,16	0,72	32,41	45,53
	1000	0,05	0,27	0,99	37,02	48,39
	2000	0,03	0,50	1,59	44,92	53,99
	5000	0,01	1,21	3,29	70,09	72,34
AAN (6.20) Mem.	200	1	0,05	0,54	28,84	135,95
	500	0,8	0,16	0,66	29,38	143,57
	1000	0,5	0,22	0,94	30,32	151,70
	2000	0,3	0,44	1,43	32,35	168,57
	5000	0,2	1,10	3,07	38,84	209,87
AAN (6.20) No Mem.	200	1	0,06	0,55	29,83	43,44
	500	0,8	0,17	0,77	31,92	46,14
	1000	0,5	0,32	1,21	35,75	50,53
	2000	0,3	0,66	1,98	43,56	59,54
	5000	0,2	1,59	4,56	68,16	88,54

Tabla 6.5: Tiempos de ejecución (en segundos) para el AAN. Distribución: dos eslabones encadenados.

#### 6.3.4.1. Dos eslabones encadenados

La Tabla 6.5 muestra los tiempos de ejecución (en segundos) para el AAN aplicado a varios MAO adaptados a la distribución formada por dos eslabones encadenados, mientras la Tabla 6.6 muestra los tiempos obtenidos con el método de Himberg. La Tabla 6.5 está dividida en cuatro secciones al considerar los dos tipos de núcleos de vecindad utilizados (según (6.18) o (6.20)), y las dos opciones en cuanto a almacenar las distancias entre los vectores de referencia en memoria (“Mem.”), o calcularlas en cada iteración sin almacenarlas (“No Mem.”). La Tabla 6.6 está dividida en dos secciones, también en función del tratamiento de dichas distancias.

Los tiempos de ejecución son similares a los obtenidos utilizando otras distribuciones (véanse las Secciones 6.3.4.2 y 6.3.4.3), aunque una mayor dimensión de los datos de entrada lógicamente involucra más operaciones y por tanto ralentiza el cálculo de las proyecciones. Esto puede ser apreciado con mayor claridad cuando el número de iteraciones de los algoritmos es elevado. La distribución

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

	r	$\theta$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
Himberg Mem.	1	1	0,11	0,88	35,81	608,41
	10	0,2	0,16	1,04	38,23	1093,95
	25	0,1	0,22	1,48	46,58	1937,60
	50	0,05	0,55	4,67	96,23	3176,72
	100	0,03	2,36	18,40	311,65	11101,60
Himberg No Mem.	1	1	0,11	0,88	36,47	92,49
	10	0,2	1,31	9,01	370,42	902,21
	25	0,1	3,30	22,96	916,65	2280,67
	50	0,05	7,58	54,81	1971,66	5209,85
	100	0,03	20,87	151,81	4605,56	14176,55

Tabla 6.6: Tiempos de ejecución (en segundos) para el método de contracción de Himberg. Distribución: dos eslabones encadenados.

formada por dos eslabones encadenados es tridimensional, aunque en la práctica ha sido definida en  $\mathbb{R}^8$ , donde sus últimos cinco componentes siempre son cero (tanto para los datos como para los vectores de referencia adaptados a ellos). De esta manera, los tiempos relativos a esta distribución pueden ser incluso más elevados que los obtenidos con la distribución formada por ocho *clusters* disjuntos con forma hipercúbica (implementados en  $\mathbb{R}^8$ ). El método de Himberg es menos sensible a este detalle de implementación. En general, para un número elevado de iteraciones su tiempo de ejecución aumenta según lo hace la dimensión “real” de los datos (dos para la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, tres para la formada por dos eslabones encadenados, y ocho para la formada por ocho *clusters* disjuntos con forma hipercúbica).

En este ejemplo el uso de memoria RAM para almacenar las distancias optimiza el proceso si el MAO no es  $(100 \times 100)$ -dimensional, excepto al aplicar el método de Himberg con más de 10 iteraciones, donde sigue siendo más rápido almacenar las distancias entre los vectores de referencia en memoria. Esto ocurre también con el experimento llevado a cabo con la distribución formada por ocho *clusters* disjuntos con forma hipercúbica (véase la Sección 6.3.4.2). No obstante, prescindir del almacenado de las distancias en memoria puede ser más rápido para MAO más pequeños. Esto sucede al aplicar el AAN para 200 y 500 iteraciones a un MAO  $(64 \times 64)$ -dimensional entrenado con la distribución formada por cuatro *clusters* de igual tamaño y diferente forma (véase la Sección 6.3.4.3). También puede suceder al aplicar el método de Himberg con una sola iteración.

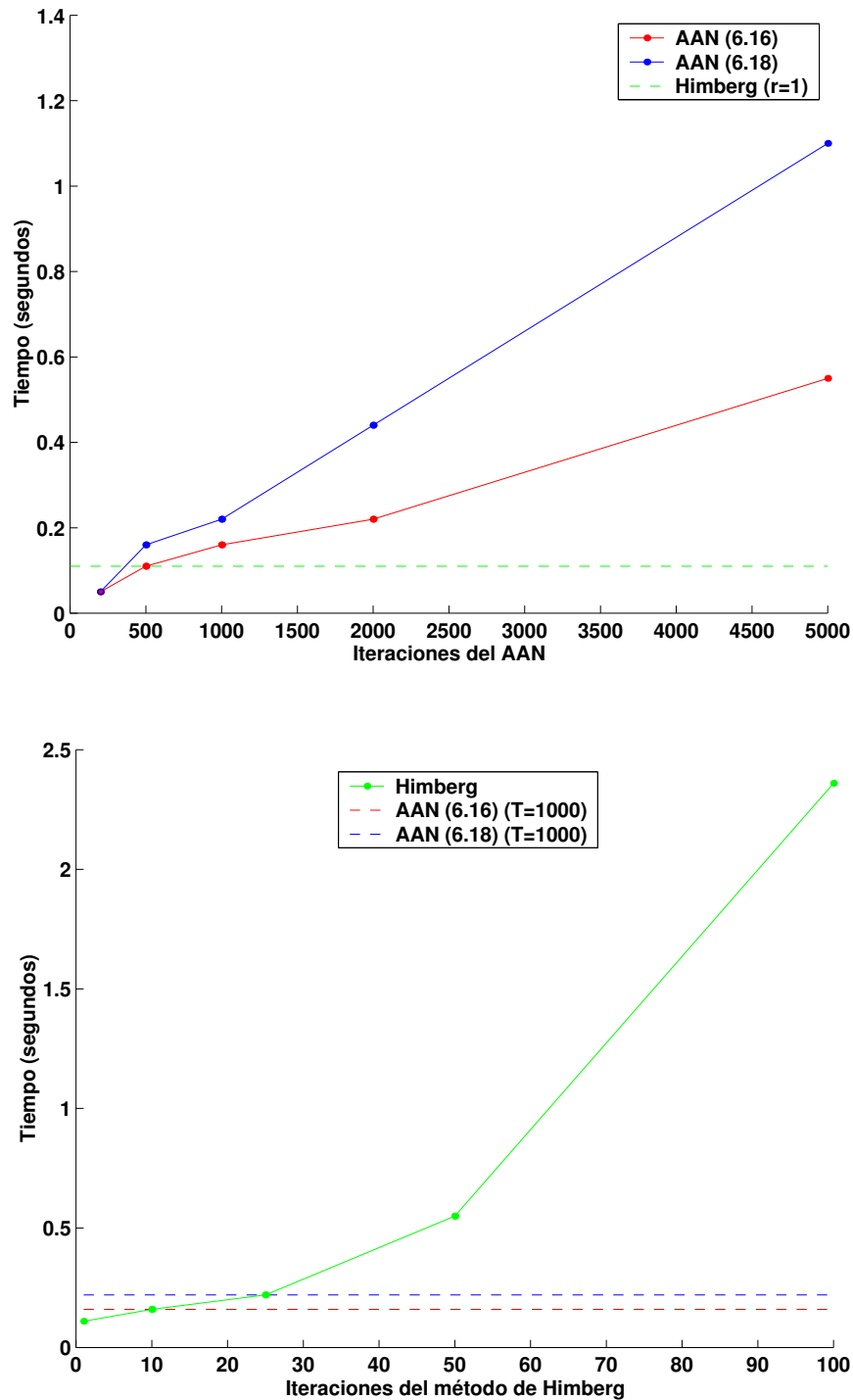


Figura 6.105: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados.

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

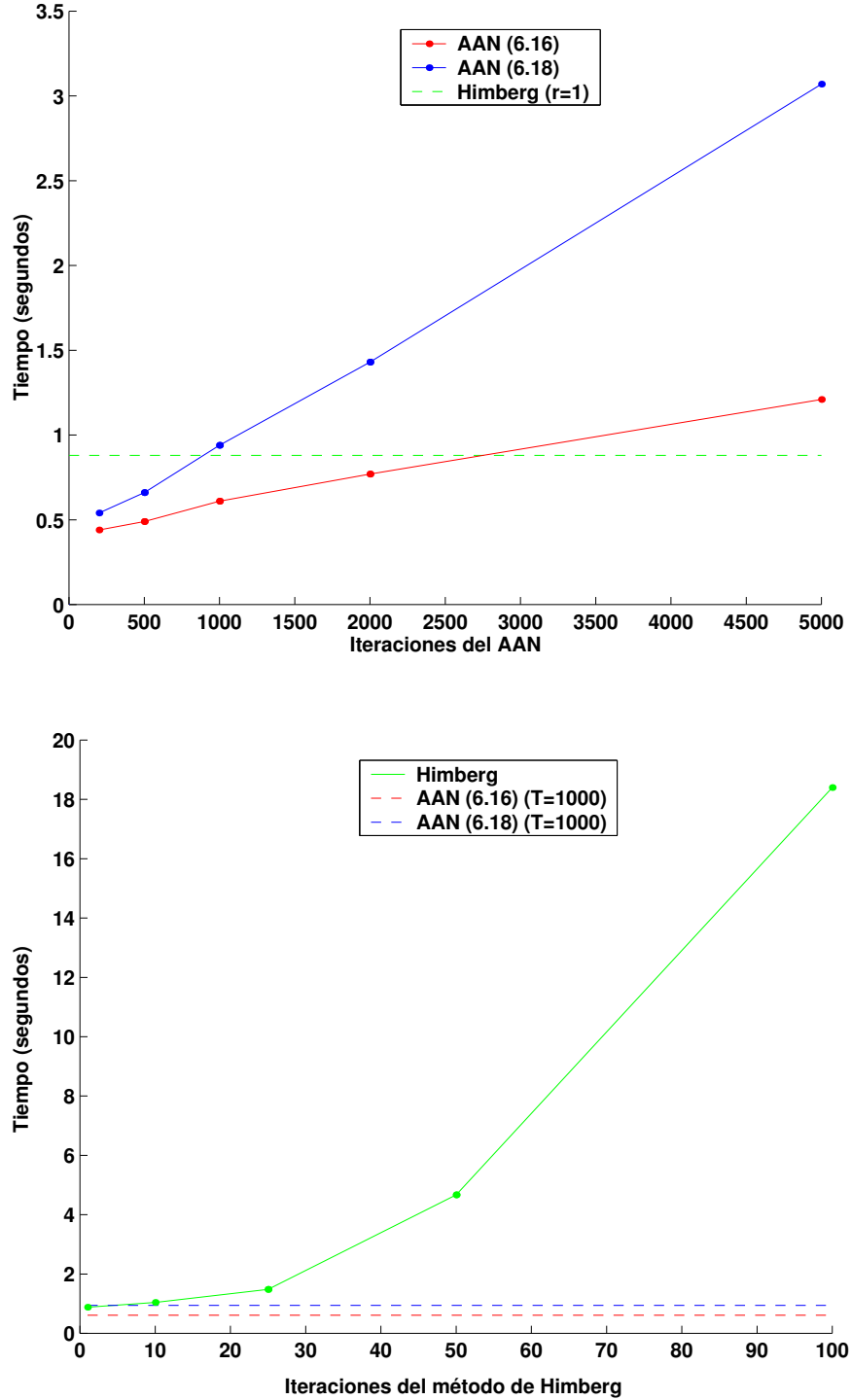


Figura 6.106: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO ( $32 \times 32$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados.

## 6. NUEVOS MÉTODOS DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS

---

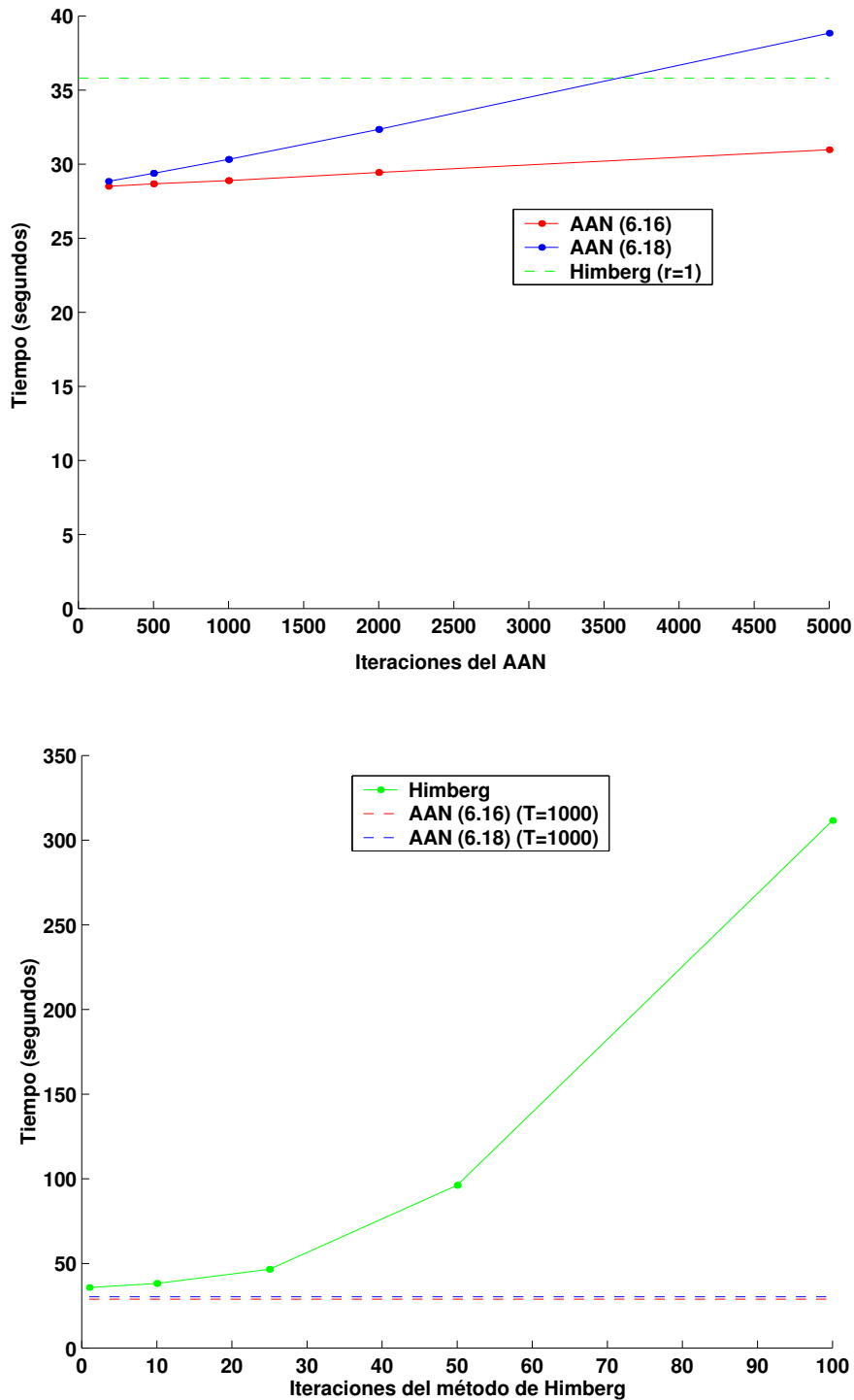


Figura 6.107: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO ( $64 \times 64$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados.

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

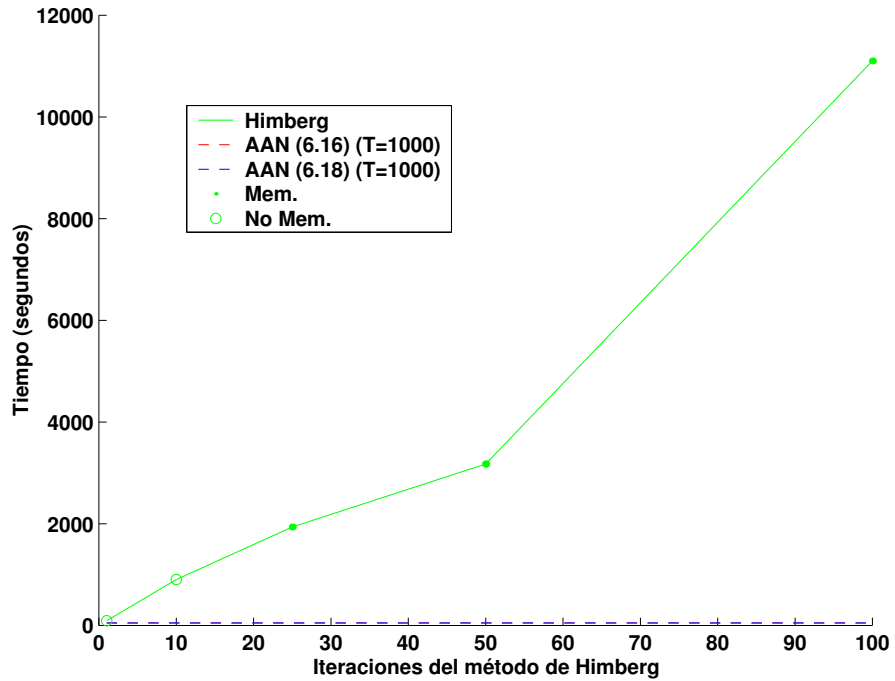
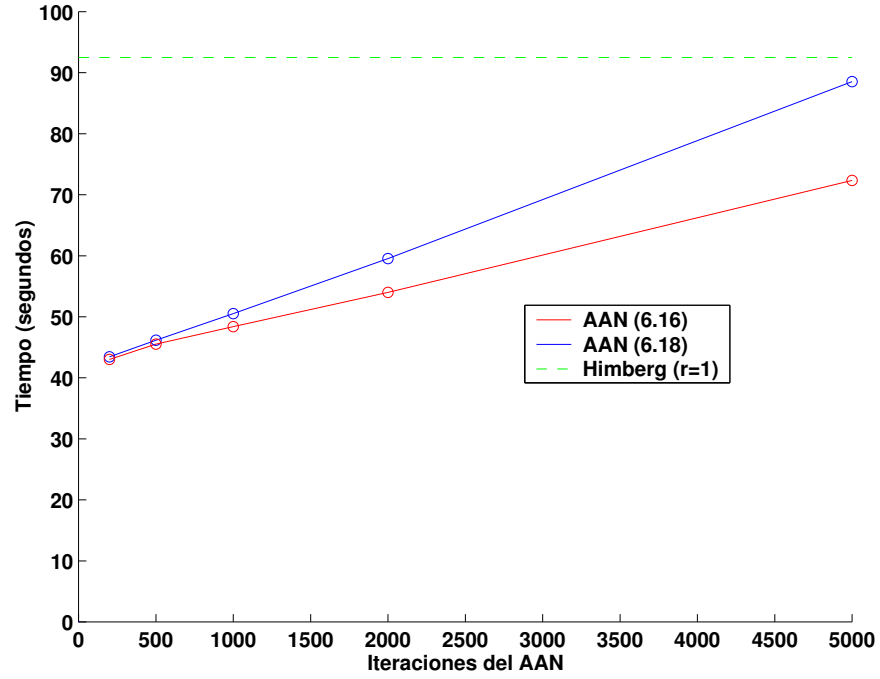


Figura 6.108: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO  $(100 \times 100)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados.

Las Figuras 6.105, 6.106, 6.107 y 6.108 muestran gráficas en las que se comparan los tiempos óptimos del AAN y del método de Himberg para MAO  $(20 \times 20)$ ,  $(32 \times 32)$ ,  $(64 \times 64)$  y  $(100 \times 100)$ -dimensionales, respectivamente, adaptados a la distribución formada por dos eslabones encadenados.

En las gráficas es posible ver como el AAN produce resultados más rápidamente con el núcleo de vecindad definido en (6.18) que en (6.20). Esto es lógico ya que el segundo define un núcleo de vecindad en el espacio de los datos y en el propio mapa, mientras que el primero lo hace únicamente en el espacio de los datos. Esto puede ser observado en el resto de experimentos llevados a cabo, véanse las Secciones 6.3.4.2 y 6.3.4.3.

El estudio sobre la complejidad computacional ( $\mathcal{O}(TM)$ ) del AAN queda confirmado en los resultados experimentales. Es posible ver como para un MAO de tamaño  $M$  fijo, los tiempos de ejecución aumentan de forma lineal con respecto al número de iteraciones  $T$  en todos los experimentos. Con respecto al método de Himberg, su complejidad computacional es  $\mathcal{O}(RM^2)$ , por lo que fijando  $M$  el tiempo de ejecución también debería aumentar de forma lineal. Esto ocurre si no se utiliza una estructura en memoria para almacenar las distancias entre vectores de referencia (véase la Tabla 6.6). Sin embargo, al emplear dicha estructura la curva del tiempo con respecto al número de iteraciones puede no ser lineal debido a factores relacionados con los aciertos de caché y fallos de página, o a efectos como el asociado a la función exponencial del lenguaje C (véase la Sección 6.3.4.2). En este caso la derivada de la curva suele ser positiva.

Las gráficas superiores de las imágenes ilustran los tiempos óptimos de ejecución del AAN frente al tiempo de ejecución del método de Himberg para una iteración, que produce la proyección más precisa. Considerando que el ANN genera proyecciones suficientemente precisas para  $T = 1000$  iteraciones, es posible apreciar (en este experimento y en el resto) como la diferencia entre el tiempo mínimo necesario para crear proyecciones precisas con el método de Himberg y el AAN aumenta a medida que lo hace el tamaño del MAO. Sin embargo, las diferencias no son significativas excepto para los MAO  $(100 \times 100)$ -dimensionales, donde éstas pueden ser superiores al 50 %. De hecho, el método de Himberg es más rápido para el MAO  $(20 \times 20)$ -dimensional. Por otro lado, para los dos MAO más pequeños las proyecciones son generadas prácticamente en tiempo real. Por estas razones el método de Himberg resulta más idóneo cuando sólo se desea analizar la proyección final y el MAO no es de tamaño muy elevado.



Cuando el objetivo es analizar el proceso dinámico de contracción de las neuronas el número de iteraciones relativas al método de Himberg debe aumentar (por ejemplo, hasta varias decenas). En estos casos existe una diferencia significativa entre los tiempos de ejecución de ambos métodos, especialmente cuando el tamaño del MAO es elevado. El uso del AAN resulta más apropiado no sólo por su mayor velocidad, sino porque las proyecciones del método de Himberg pierden precisión a medida que aumenta el número de iteraciones. El método de Himberg puede ser aplicado a MAO de tamaño medio ya que sólo tarda unos pocos segundos en generar las sucesivas proyecciones, sin embargo, resulta ser demasiado lento para tratar MAO de tamaño elevado.

La Figura 6.109 muestra varias proyecciones de los métodos de contracción aplicados a un MAO ( $100 \times 100$ )-dimensional adaptado a la distribución formada por dos eslabones encadenados. El objetivo de esta figura es ilustrar que incluso con MAO de gran tamaño el AAN es capaz de generar proyecciones similares a la más precisa del método de Himberg, utilizando un número de iteraciones relativamente bajo. La figura muestra la proyección de Himberg para  $r = 1$  y  $T = 1$  (a); la proyección del AAN utilizando el núcleo de vecindad definido en (6.20), y el conjunto de parámetros “estándar”, con  $T = 500$  y  $T = 1000$  en (b) y (c), respectivamente. La proyección de (c) es ligeramente más precisa que la de (b), aunque ambas aportan prácticamente la misma información que la proyección de Himberg. Finalmente la imagen (d) ilustra la proyección obtenida con el método coordenadas adaptativas, la cual es muy diferente a las anteriores y no describe adecuadamente las características de la distribución.

#### 6.3.4.2. Ocho *clusters* disjuntos con forma hipercúbica

La Tabla 6.7 muestra los tiempos de ejecución (en segundos) para el AAN aplicado a varios MAO adaptados a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica, mientras la Tabla 6.8 muestra los tiempos obtenidos con el método de Himberg. El análisis de los tiempos de ejecución aporta la misma información que el estudio llevado con la distribución formada por dos eslabones encadenados, ayudando a corroborar las conclusiones.

Lo más destacable del experimento es el tiempo de ejecución del método de Himberg (con almacenamiento de distancias en memoria) para  $r = 50$  y  $\theta = 0,05$ , que es igual a 361,14s. Éste resulta ser mucho mayor que el empleado para el doble de iteraciones con  $\theta = 0,03$  (180,65s). Por otro lado, el algoritmo tarda 102,60s

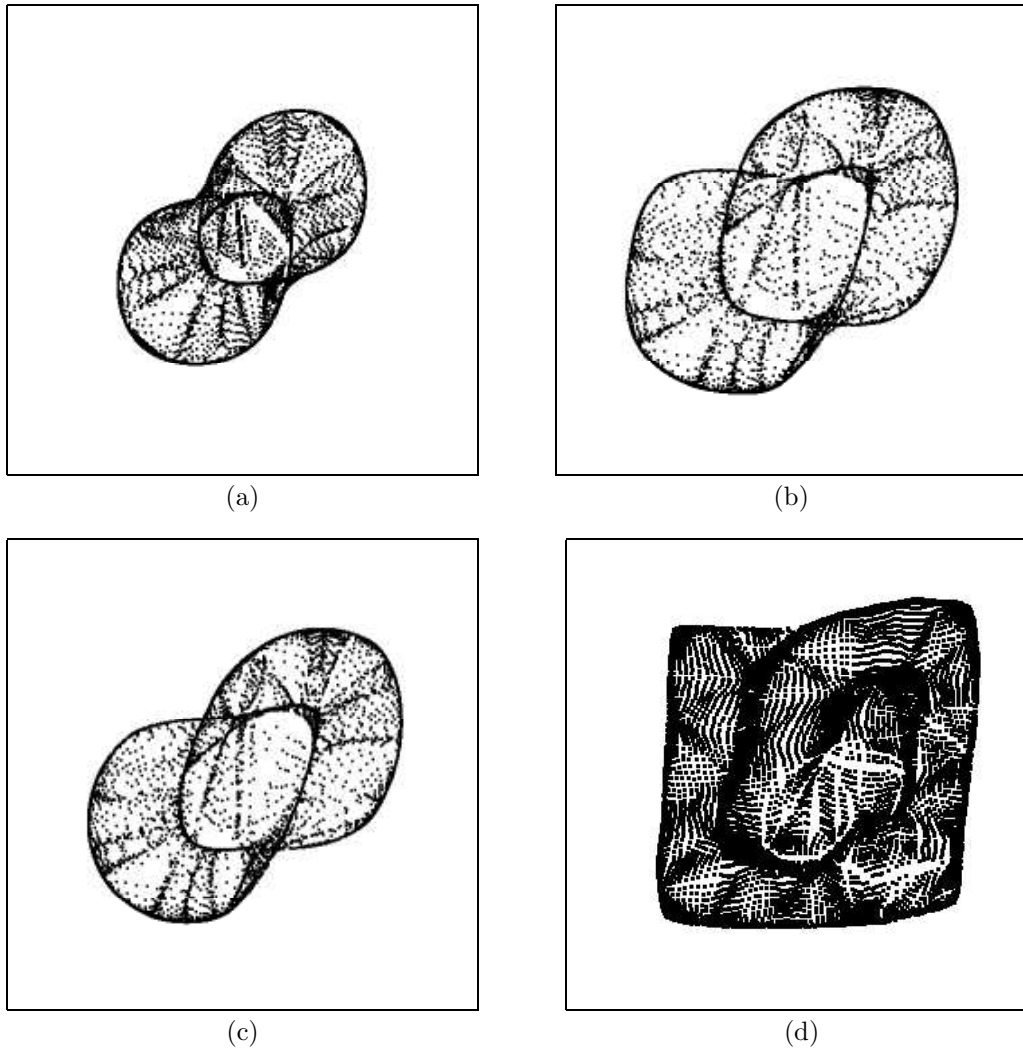


Figura 6.109: Proyecciones de los métodos de contracción para un MAO  $(100 \times 100)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados. Método de Himberg con  $r = 1$  (a). AAN con  $T = 500$  (b) y  $T = 1000$  (c). Coordenadas adaptativas (d).

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

	$T$	$\lambda(0)$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
AAN (6.18) Mem.	200	0,2	0,05	0,22	3,35	238,16
	500	0,1	0,11	0,28	3,46	200,14
	1000	0,05	0,16	0,38	3,73	201,69
	2000	0,03	0,22	0,50	4,18	204,87
	5000	0,01	0,49	0,99	5,44	238,43
AAN (6.18) No Mem.	200	0,2	0,05	0,21	3,51	42,07
	500	0,1	0,11	0,33	3,90	43,72
	1000	0,05	0,22	0,50	4,51	46,58
	2000	0,03	0,38	0,77	5,71	52,23
	5000	0,01	0,94	1,70	9,28	69,10
AAN (6.20) Mem.	200	1	0,11	0,28	3,79	191,25
	500	0,8	0,16	0,44	4,23	193,61
	1000	0,5	0,27	0,66	5,11	197,73
	2000	0,3	0,38	1,26	6,92	202,84
	5000	0,2	0,99	2,75	12,57	233,82
AAN (6.20) No Mem.	200	1	0,11	0,33	3,84	42,51
	500	0,8	0,16	0,49	4,62	44,76
	1000	0,5	0,27	0,82	5,95	49,11
	2000	0,3	0,60	1,43	8,46	57,50
	5000	0,2	1,48	3,46	16,59	82,67

Tabla 6.7: Tiempos de ejecución (en segundos) para el AAN. Distribución: ocho *clusters* disjuntos con forma hipercúbica.

	$r$	$\theta$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
Himberg Mem.	1	1	0,11	0,66	10,71	714,41
	10	0,2	0,16	0,82	13,01	1268,01
	25	0,1	0,22	2,75	63,27	2394,47
	50	0,05	2,58	11,15	361,14	4950,71
	100	0,03	3,13	26,47	180,65	12379,17
Himberg No Mem.	1	1	0,11	0,72	10,71	87,22
	10	0,2	1,37	7,52	107,05	878,21
	25	0,1	3,57	22,52	422,04	2849,48
	50	0,05	14,17	73,27	1823,86	10204,33
	100	0,03	28,07	171,09	2398,87	17324,60

Tabla 6.8: Tiempos de ejecución (en segundos) para el método de contracción de Himberg. Distribución: ocho *clusters* disjuntos con forma hipercúbica.

en calcular la proyección para  $r = 50$  y  $\theta = 0,03$ . Para un MAO determinado, la velocidad del método de Himberg sólo depende del número de iteraciones  $r$  empleadas, al menos teóricamente. Sin embargo, en la práctica puede depender también de la escala en la que estén definidos los datos (que influye en las distancias entre los vectores de referencia), y en el valor del parámetro  $\theta$ . Tras realizar un análisis del código fuente y estructuras de datos involucradas la respuesta a este problema parece radicar en el valor de retorno de la función  $y = e^x$ , es decir, la función “double exp(double x)” de la librería estándar de C “math.h”. Para valores negativos elevados de su argumento la función puede retornar cero o un valor muy próximo a éste. El tratamiento del cero puede acelerar el algoritmo ya que ciertas operaciones pueden ser “ignoradas”, mientras que el distinto de cero debe ser procesado siempre. De hecho, tras implementar una modificación (operación condicional) para convertir dichos valores ligeramente positivos (inferiores a  $10^{-4}$ ) de la matriz  $\tilde{S}$  a cero, el tiempo de ejecución fue reducido a 35,15s para  $r = 50$  y  $\theta = 0,05$ , lo que supone una reducción superior al 90 %. En cualquier caso, esta modificación no ha sido tomada en cuenta para calcular los tiempos de ejecución por varios motivos: En primer lugar, supone variar el algoritmo original de Himberg. En segundo, optimizaciones similares no han sido tomadas en cuenta en la implementación del ANN. Por último, los resultados no siempre aumentan la velocidad del algoritmo. Nótese que el valor de retorno de la función exponencial depende de distancias entre vectores de referencia y del parámetro  $\theta$  (véase (5.14)). Cuanto mayor es  $\theta$  y menor la distancia, mayor es el valor de retorno, lo cual ayuda a evitar el problema anterior. Los tiempos de ejecución para la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, donde  $\theta$  toma valores elevados (ya que las distancias entre los datos son mayores), son mayores al implementar la variante al tener que ejecutar la operación condicional.

Por otro lado, la velocidad del ANN depende de numerosos factores: el número de iteraciones, el núcleo de vecindad, la función decreciente del radio de vecindad, la función decreciente de la tasa de aproximación, y el orden de las neuronas elegidas. En los experimentos todos estos elementos quedan fijados inicialmente, excepto el orden de las neuronas elegidas que influye en el número medio de neuronas que son adaptadas en cada iteración. No obstante, en la práctica este orden apenas afecta al tiempo de ejecución ya que el número de iteraciones suele ser lo suficientemente elevado como para que dicha media no varíe de forma significativa.

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

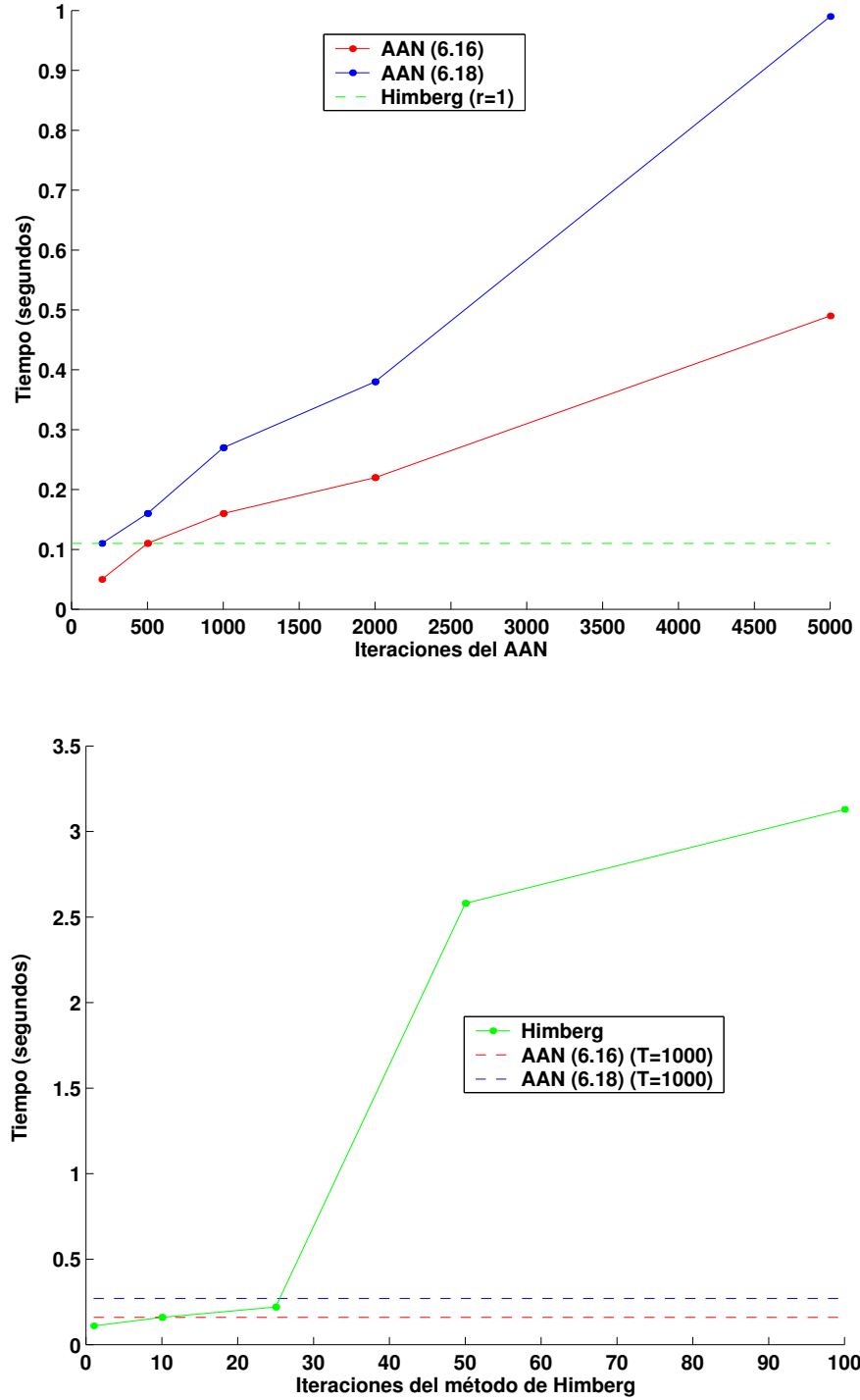


Figura 6.110: Comparación de tiempos óptimos de ejecución del AAN y del método de Himberg para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica.

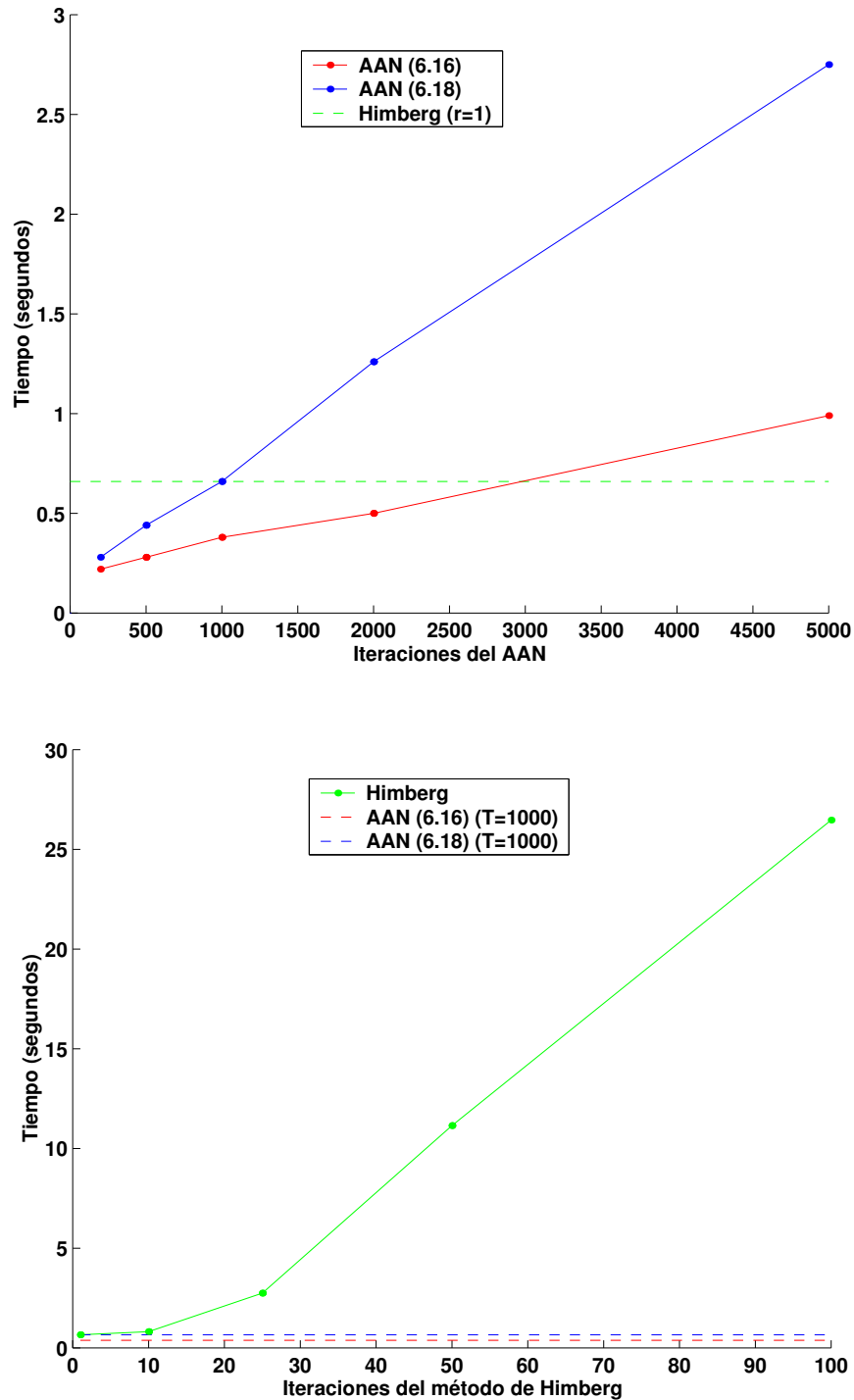


Figura 6.111: Comparación de tiempos óptimos de ejecución del AAN y del método de Himberg para un MAO ( $32 \times 32$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica.

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

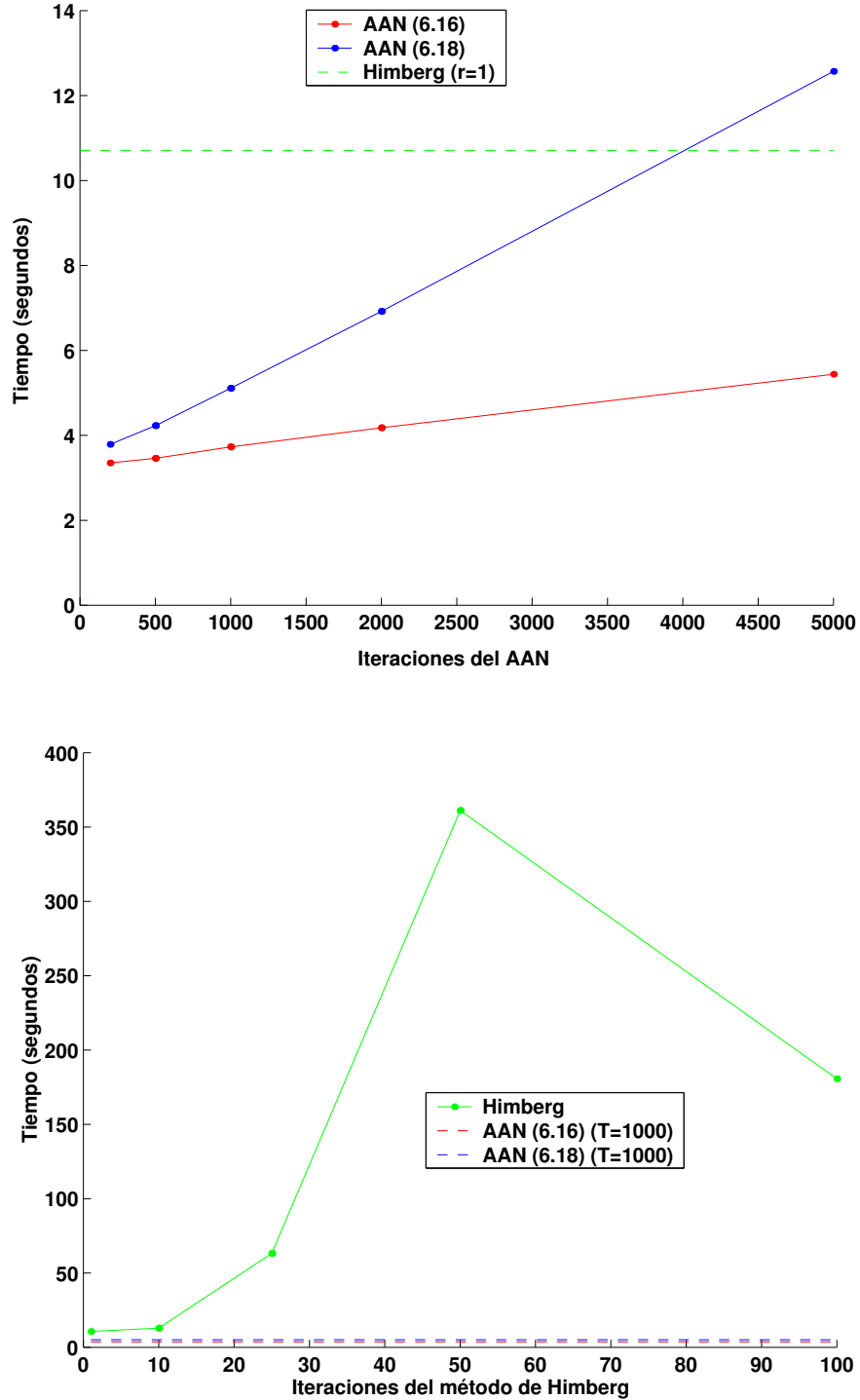


Figura 6.112: Comparación de tiempos óptimos de ejecución del AAN y del método de Himberg para un MAO ( $64 \times 64$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica.

## 6. NUEVOS MÉTODOS DE VISUALIZACIÓN DE MAPAS AUTO-ORGANIZATIVOS

---

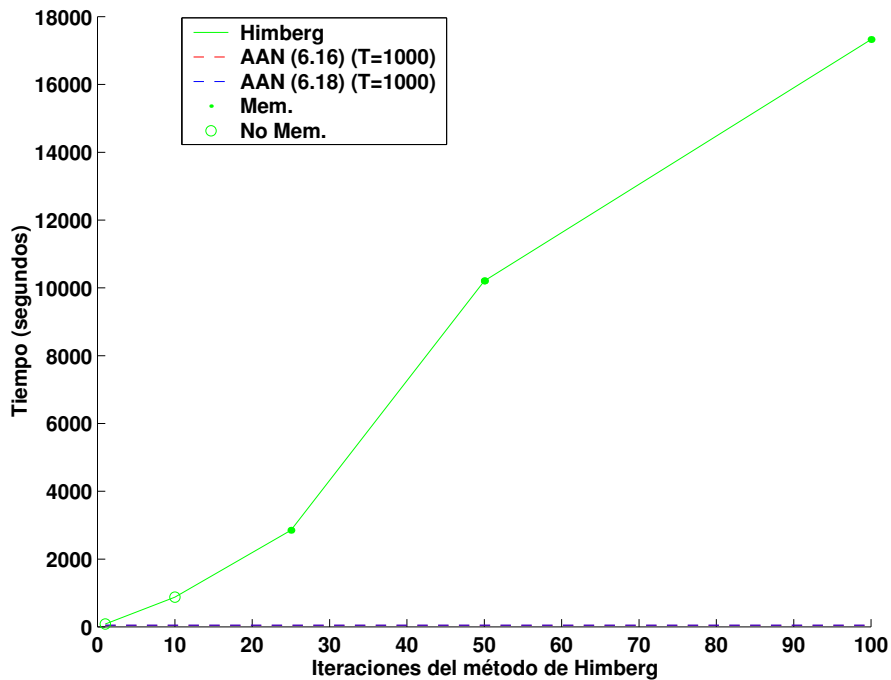
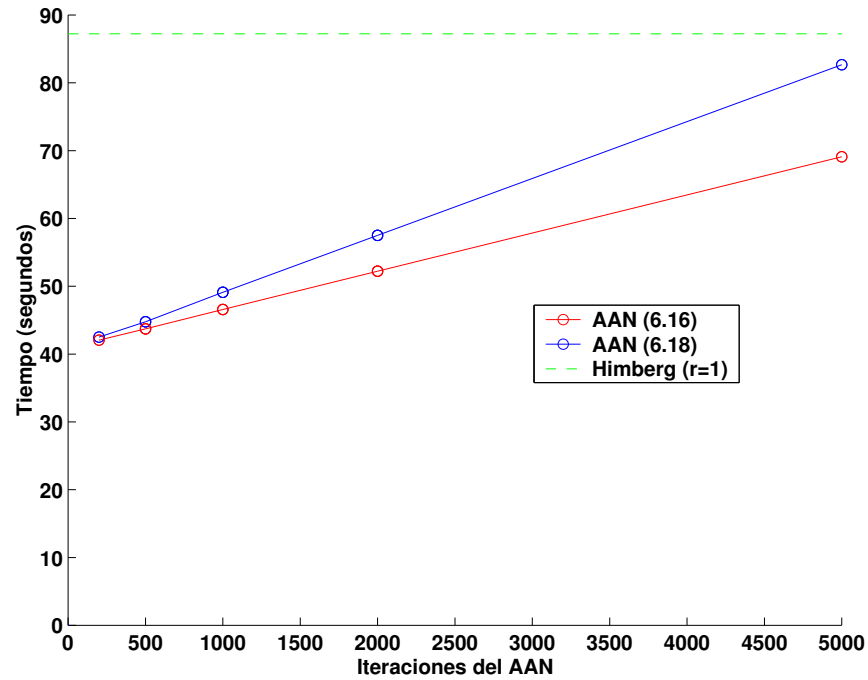


Figura 6.113: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO (100 × 100)-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica.



### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

	$T$	$\lambda(0)$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
AAN (6.18) Mem.	200	0,1	0,05	0,11	1,64	159,78
	500	0,05	0,05	0,17	1,87	163,97
	1000	0,02	0,11	0,33	2,14	168,79
	2000	0,01	0,22	0,44	2,74	196,63
	5000	0,005	0,55	0,99	4,62	228,93
(6.18) AAN No Mem.	200	0,1	0,06	0,11	1,42	24,16
	500	0,05	0,11	0,22	1,76	25,43
	1000	0,02	0,17	0,28	2,25	27,58
	2000	0,01	0,38	0,49	3,14	31,85
	5000	0,005	0,82	1,26	6,04	44,60
AAN (6.20) Mem.	200	0,8	0,05	0,17	1,93	116,06
	500	0,3	0,11	0,32	2,47	121,16
	1000	0,1	0,22	0,55	3,41	129,02
	2000	0,05	0,44	0,99	5,16	146,27
	5000	0,02	0,99	2,36	10,66	195,37
AAN (6.20) No Mem.	200	0,8	0,11	0,17	1,76	24,60
	500	0,3	0,17	0,33	2,41	26,64
	1000	0,1	0,27	0,60	3,57	29,99
	2000	0,05	0,55	1,21	5,88	36,97
	5000	0,02	1,48	3,07	13,51	59,43

Tabla 6.9: Tiempos de ejecución (en segundos) para el AAN. Distribución: cuatro *clusters* de igual tamaño y diferente forma.

Las Figuras 6.110, 6.111, 6.112 y 6.113 muestran gráficas en las que se comparan los tiempos óptimos del AAN y del método de Himberg para MAO  $(20 \times 20)$ ,  $(32 \times 32)$ ,  $(64 \times 64)$  y  $(100 \times 100)$ -dimensionales, respectivamente, adaptados a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica. Las gráficas son similares a las presentadas en el apartado anterior, aportando la misma información. Destaca el elevado tiempo de ejecución del algoritmo de Himberg para  $r = 50$ .

#### 6.3.4.3. Cuatro *clusters* de igual tamaño y diferente forma

La Tabla 6.9 muestra los tiempos de ejecución (en segundos) para el AAN aplicado a varios MAO adaptados a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, mientras la Tabla 6.10 muestra los tiempos obtenidos con el método de Himberg. El análisis de los tiempos de ejecución aporta la misma información que los estudios llevados a cabo en los apartados anteriores.

	$r$	$\theta$	$(20 \times 20)$	$(32 \times 32)$	$(64 \times 64)$	$(100 \times 100)$
Himberg Mem.	1	1000	0.06	0.55	9.06	371.25
	10	100	0.55	3.68	57.45	951.04
	25	50	0.66	3.79	57.89	1559.28
	50	25	0.66	4.17	69.38	2376.03
	100	15	1.15	6.87	106.39	5300.86
Himberg No Mem.	1	1000	0.06	0.56	10.43	77.56
	10	100	0.83	3.96	249.47	1572.46
	25	50	1.92	9.34	554.58	1254.12
	50	25	3.73	17.63	1000.90	6475.60
	100	15	7.20	34.61	1585.15	10682.46

Tabla 6.10: Tiempos de ejecución (en segundos) para el método de contracción de Himberg. Distribución: cuatro *clusters* de igual tamaño y diferente forma.

En este ejemplo destaca el tiempo empleado por el AAN para 200 y 500 iteraciones cuando el MAO es  $(64 \times 64)$ -dimensional, que resulta ser más bajo cuando las distancias entre los vectores de referencia no son almacenadas en memoria. Por otro lado, la escala original de los datos obliga a escoger valores elevados para el parámetro  $\theta$  en el algoritmo de Himberg. Esto ayuda a evitar los efectos de la función exponencial (véase la Sección 6.3.4.2), por lo que la variante diseñada para igualar a cero componentes muy pequeños de la matriz  $\tilde{S}$  no consigue reducir el tiempo de ejecución del algoritmo.

Las Figuras 6.114, 6.115, 6.116 y 6.117 muestran gráficas en las que se comparan los tiempos óptimos del AAN y del método de Himberg para MAO  $(20 \times 20)$ ,  $(32 \times 32)$ ,  $(64 \times 64)$  y  $(100 \times 100)$ -dimensionales, respectivamente, adaptados a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma. Las gráficas son similares a las presentadas en los apartados anteriores, aportando la misma información.

### 6.3.5. Discusión

Los métodos de contracción mejoran la visualización y el análisis de *clusters* del MAO. Representan una seria alternativa a los métodos tradicionales de análisis de *clusters* ya que aportan más información útil (véase la Sección 6.3.1). Aunque existen variantes de los métodos tradicionales que pueden generar imágenes donde cada *cluster* aparece con un color diferente (superposición de planos de componentes y matriz de distancias basada en la suma de todas las distancias, véanse las Secciones 5.3.2 y 5.1.4), estas imágenes son monocromáticas y no

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

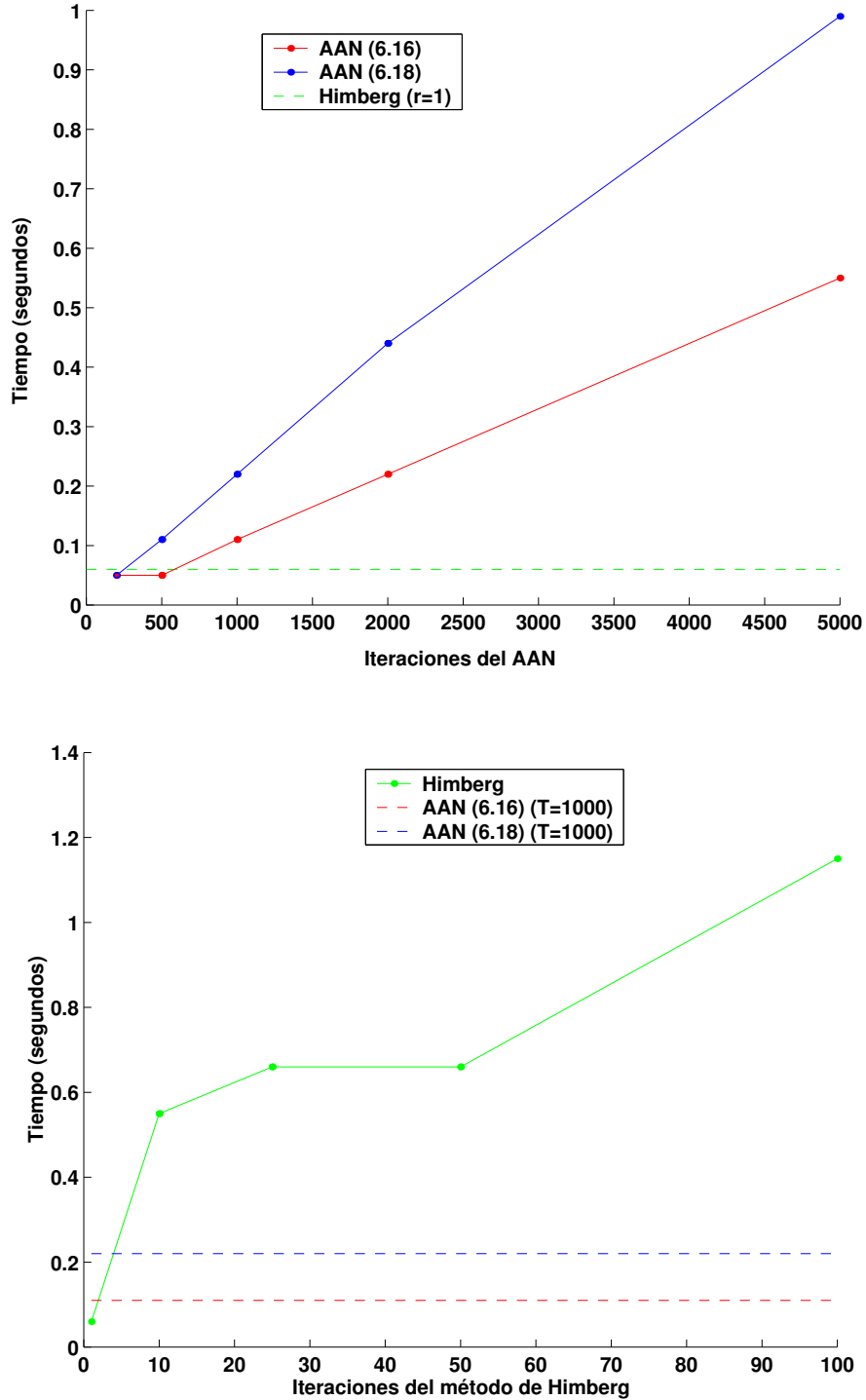


Figura 6.114: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

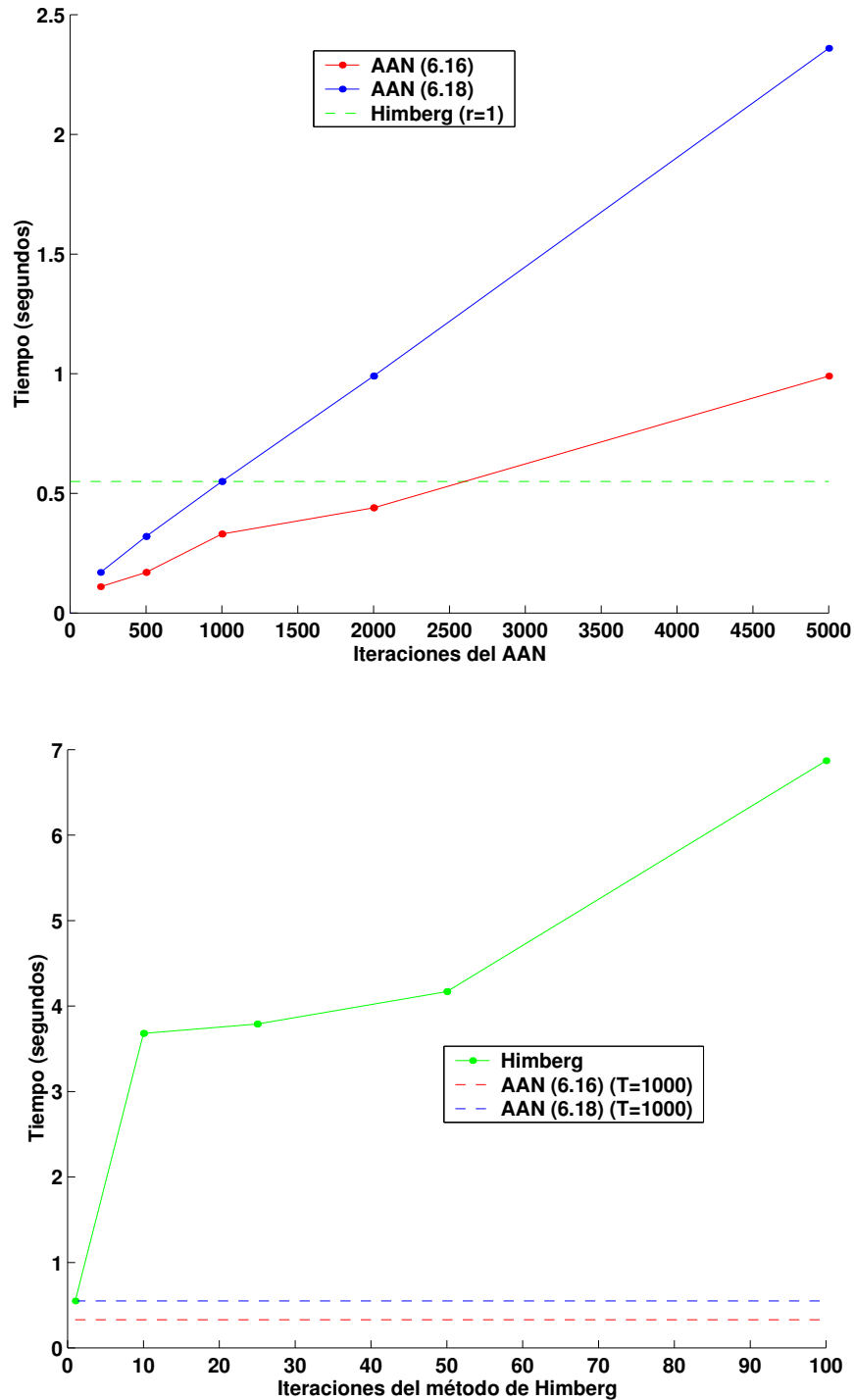


Figura 6.115: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO  $(32 \times 32)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

### 6.3. ALGORITMO DE AGRUPACIÓN DE NEURONAS

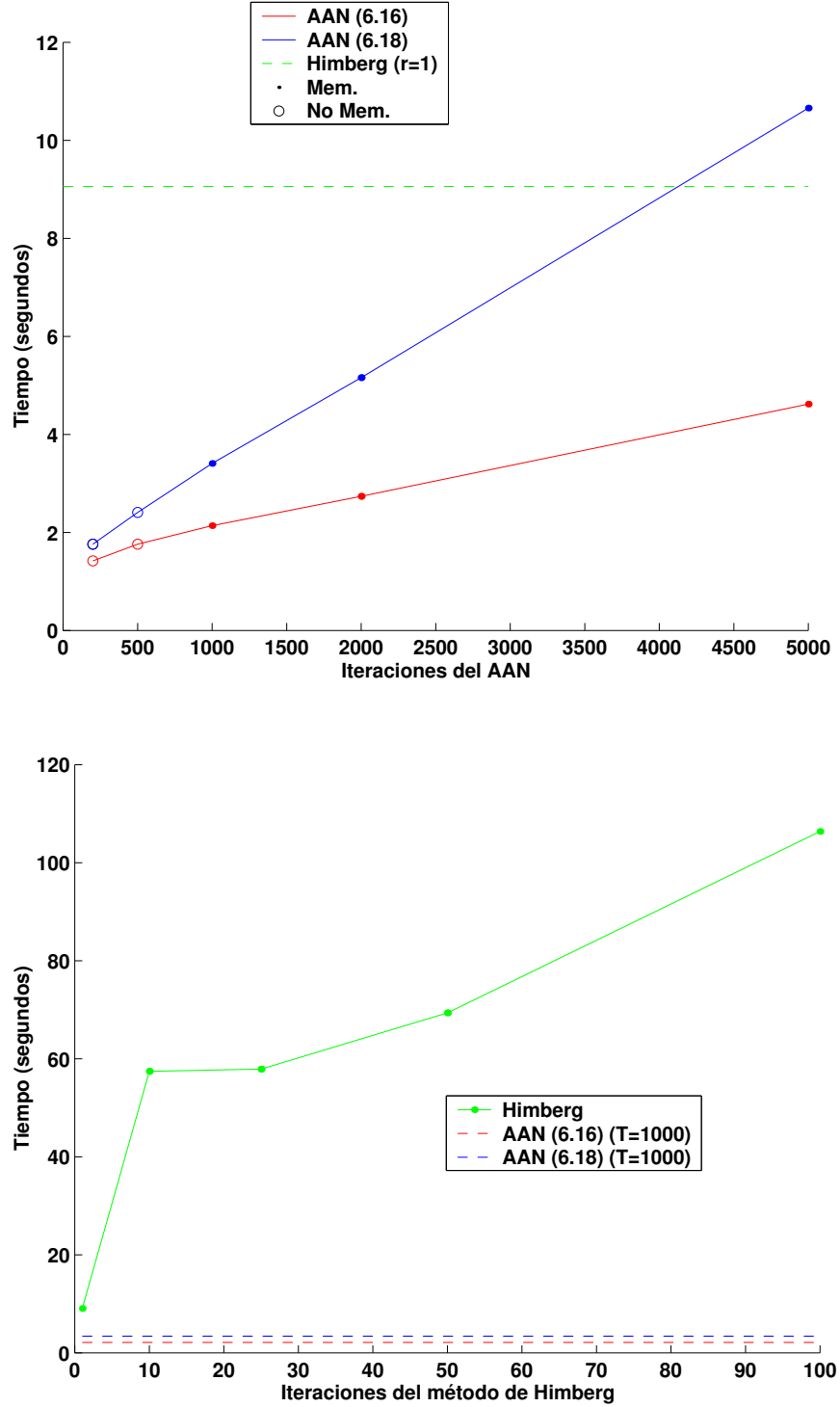


Figura 6.116: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO ( $64 \times 64$ )-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

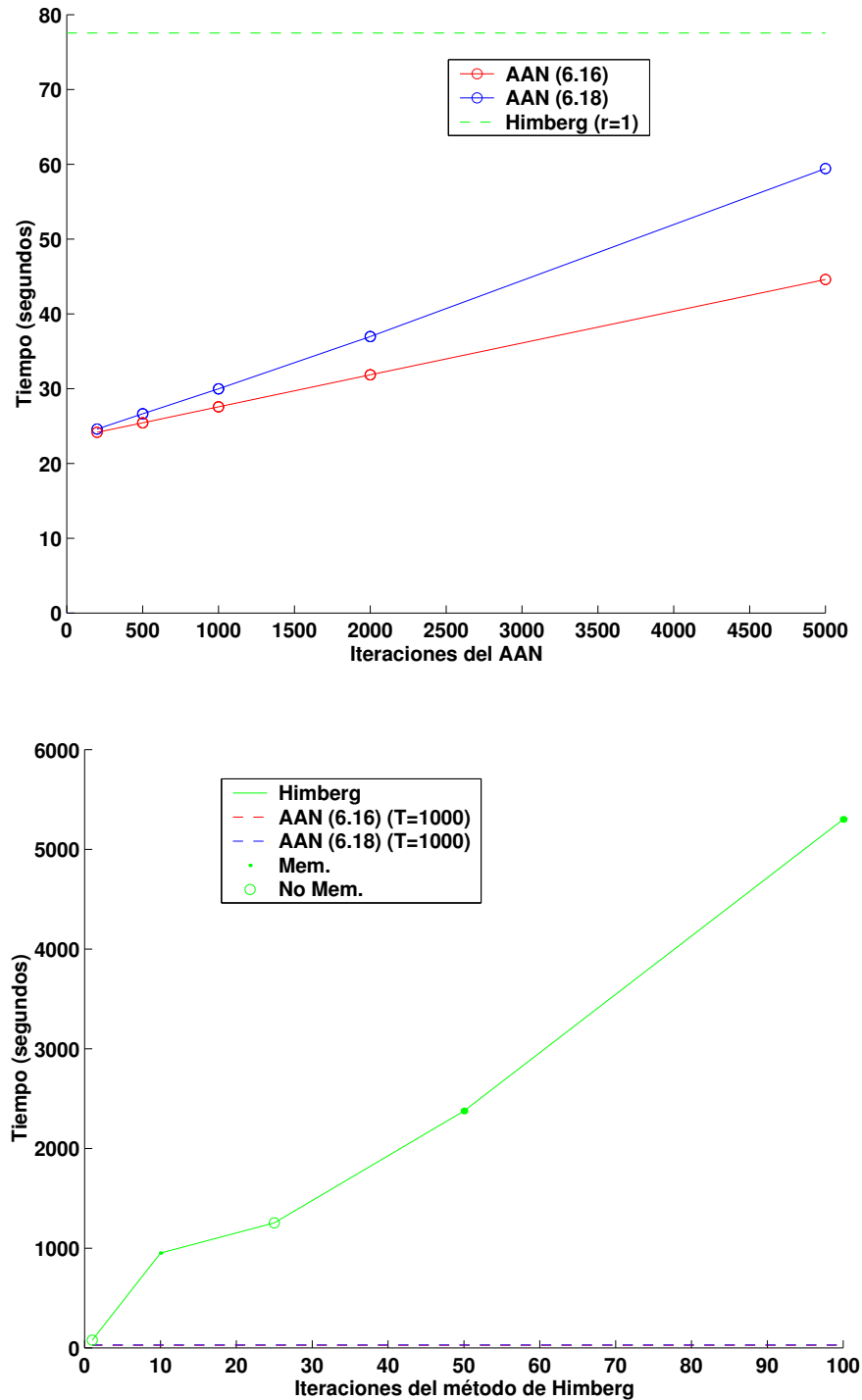


Figura 6.117: Comparación de tiempos de ejecución óptimos del AAN y del método de Himberg para un MAO ( $100 \times 100$ )-dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma.

suelen generar descripciones adecuadas con distribuciones de datos reales, por lo que las prestaciones de los métodos de contracción son claramente superiores.

El AAN es una técnica de visualización rápida y potente para el análisis de estructuras en datos de dimensión elevada, y puede proporcionar una ayuda importante en tareas difíciles de clasificación. Comparte varias características del modelo de contracción de Himberg (véase la Sección 5.5.2.3), entre las que destaca la producción de una visualización “suave” de los *clusters* utilizando una heurística razonable. El algoritmo aporta una proyección no-lineal de los vectores de referencia, donde es posible detectar *clusters* y otro tipo de estructuras. La proyección puede ser encadenada a la original regular mediante un código de colores para generar una imagen donde cada *cluster* aparece con un color diferente. Esto facilita el etiquetado del MAO y el análisis de trayectorias, aporta una idea sobre las distancias entre los *clusters* e incluso puede ayudar a detectar discontinuidades en el MAO, por lo que también sirve como una medida de bondad. Con respecto a los códigos de colores empleados con MAO cuadrados, el plano  $B = 0$  del cubo RGB parece producir descripciones más adecuadas que el plano  $B = 255 - G$ .

El método de Himberg produce descripciones (proyecciones) más precisas que el AAN ya que la naturaleza aleatoria de este último produce ligeras distorsiones en sus proyecciones. La ventaja sobre el método de Himberg es el aumento de velocidad cuando el tamaño del MAO es elevado, sin que esto repercuta significativamente en la calidad de la imagen final (las imágenes generadas siguen siendo similares a las obtenidas con el método de Himberg). Nótese que las proyecciones generadas a partir de los métodos de contracción deben contener un número de puntos o neuronas suficiente como para aportar información útil, por lo que los MAO utilizados deben ser de tamaño medio o elevado (más de unos pocos cientos de neuronas).

La complejidad computacional del método de Himberg se acerca a  $\mathcal{O}(RM^2)$ , donde  $R$  es el número de iteraciones y  $M$  el número de neuronas del MAO, mientras que la del AAN se aproxima a  $\mathcal{O}(M^{3/2})$ . A pesar de que los cálculos relativos al método de Himberg son más sencillos, el AAN resulta más adecuado a medida que el tamaño del MAO aumenta, y especialmente si se desea observar el proceso dinámico de contracción, por razones exploratorias. Nótese que la complejidad del método de Himberg es  $\mathcal{O}(M^2)$  ( $r = 1$ ) si sólo se desea ver la proyección final. Sin embargo, si el objetivo consiste en analizar el proceso dinámico de contracción el

número de iteraciones  $r$  tendría que tomar un valor más elevado, lo cual puede alargar considerablemente su tiempo de ejecución.

En la práctica, estos métodos no producen resultados interesantes con MAO pequeños (aproximadamente menos de cien neuronas). Para los de tamaño medio, el método de Himberg produce los resultados más precisos prácticamente en tiempo real. Por tanto, el algoritmo propuesto en este apartado debe ser aplicado a MAO de tamaño relativamente elevado (más de mil neuronas), sobre todo si el objetivo es analizar el proceso de contracción, dada su complejidad computacional más ligera. Por otro lado, el algoritmo no está limitado a MAO bidimensionales.

La naturaleza aleatoria del algoritmo propuesto permite generar diferentes proyecciones no-lineales cada vez que es ejecutado (ya que produce diferentes secuencias de neuronas “elegidas”), lo cual puede ser interesante desde un punto de vista exploratorio. Por otro lado, si se desea analizar el proceso dinámico de contracción el analista debe poder visualizarlo repetidas veces. En este caso sólo es necesario almacenar la secuencia de neuronas elegidas en cada iteración: una lista de números enteros cuyo tamaño es el número de iteraciones  $T$  del algoritmo.

La diferencia principal entre el MST y el AAN reside en que el primero proyecta patrones de la propia distribución de datos, mientras que el segundo proyecta un conjunto de centroides o vectores de referencia adaptados a los datos. De esta manera, el MST generalmente produce imágenes de mayor resolución, lo cual puede ser importante desde un punto de vista exploratorio. En cualquier caso, ambos métodos pueden generar imágenes similares cuando el MAO contiene un número elevado de neuronas. Sin embargo, en esos casos el MST necesita utilizar un número elevado de pares de neuronas para generar proyecciones precisas, por lo que resulta mucho más lento que el AAN.

Finalmente, destaca la posibilidad de modificar el método de Himberg para reducir su complejidad computacional. Aunque varias ideas han sido propuestas, éstas todavía no han sido implementadas, por lo que su análisis es incluido como una de las líneas futuras de esta tesis.

## 6.4. Proyección de Grafos

Este apartado presenta una estrategia muy diferente, no utilizada hasta la fecha, para analizar MAO de manera visual. Consiste en la visualización explícita en el mapa o espacio de observación de grafos que unen neuronas cuyos vectores de



referencia se hallan próximos en el espacio de los datos, como son el grafo Hebbiano (“triangulación inducida de Delaunay”), el árbol generador mínimo (AGM), o grafos que conectan cada neurona  $\eta_i$  con el conjunto de neuronas asociadas a los  $K$  primeros vectores de referencia más cercanos al de  $\eta_i$ .

El objetivo consiste en generar una medida intuitiva y visual de la preservación de la topología en cada zona del MAO, que aporte información sobre la calidad del proceso de adaptación del MAO y sobre la posible dimensión intrínseca de los datos en su dominio. La estrategia propuesta pretende localizar las posibles discontinuidades en el MAO que contribuyen a una pobre preservación de la topología, en vez de aportar únicamente un valor cuantitativo mediante una medida de bondad. Las imágenes generadas permiten observar áreas del MAO que no han sido adaptadas a la distribución de forma ordenada, indicando dónde la dimensión intrínseca de los datos es distinta de la dimensión del MAO. Como los grafos conectan vectores de referencia próximos entre sí, se espera que sus aristas proyectadas conecten neuronas vecinas en el mapa. En otras palabras, las aristas proyectadas deben ser cortas donde el MAO preserva adecuadamente la topología y largas donde existen discontinuidades. Cuando existen discontinuidades es posible examinar la proporción de aristas cortas para determinar si el MAO se ha adaptado a los datos “lo mejor posible”, aunque su dimensión no coincida con la intrínseca de los datos. Por otro lado, las imágenes también pueden indicar la existencia de *clusters* y bordes entre éstos.

##### 6.4.1. Descripción del problema

En la práctica, dada la naturaleza aleatoria y los diferentes parámetros de entrenamiento asociados al algoritmo secuencial de entrenamiento del MAO, varios son entrenados y comparados con diversas medidas de bondad en un intento por encontrar uno óptimo [Koh01]. Un criterio usado frecuentemente es el error de cuantización medio que mide la precisión estadística del MAO, es decir, lo bien que se han adaptado los vectores de referencia a los datos de la distribución de entrada. Sin embargo, estos vectores de referencia deben aparecer ordenados de manera que neuronas vecinas en el mapa representen patrones de entrada similares. En otras palabras, la precisión con la que se preserva la topología o las relaciones de vecindad en el espacio de los datos debe ser medida también. Varias medidas de preservación de la topología son descritas en la Sección 3.2.7.5. Algunas están basadas en la idea de que los dos vectores de referencia más próximos

a un patrón de entrada deben corresponder a neuronas vecinas en el mapa. El error topográfico [Kiv96] calcula la proporción de patrones de entrada con los cuales no se satisface el supuesto anterior. La medida Hebbiana [Pol97] compara el grafo generado al conectar los vectores de referencia de neuronas adyacentes con el grafo Hebbiano, creado con el principio de aprendizaje Hebbiano competitivo [Mar93], donde dos vectores son conectados si son los dos más cercanos a un patrón de entrada. Estas medidas ayudan a determinar si la dimensión del MAO es apropiada para tratar la distribución y para visualizarla.

Sin embargo, varios subconjuntos de la distribución de datos pueden presentar una dimensión intrínseca diferente. Es decir, ésta no siempre es constante en todo el dominio de los datos, por lo que varias zonas de un MAO pueden adaptarse a ellos de manera distinta, con menor o mayor grado de preservación de la topología. Por esta razón no siempre es aconsejable medirla mediante un solo valor cuantitativo. Aunque el analista puede dividir el espacio de entrada en varias regiones para estudiarla en cada una de ellas, dicha subdivisión no es trivial. El analista puede aplicar algoritmos de CV y métodos de proyección de datos multidimensionales para conseguir aislar las diversas clases que puedan tener una dimensión intrínseca diferente. Afortunadamente los MAO realizan ambas tareas, por lo que el propio MAO puede ser utilizado para estudiar el grado de preservación de la topología y la dimensión intrínseca de los datos, en cada región de la distribución.

No obstante, en general, los métodos tradicionales de visualización de MAO no aportan información relativa al grado de preservación de la topología del MAO, ni sobre la dimensión intrínseca de los datos. Para poder visualizar estructuras de menor dimensión intrínseca que la propia dimensión del MAO en el mapa es posible recurrir a histogramas de datos cuando el tamaño del MAO es elevado (véase la Figura 6.47), o a métodos de proyección como el MST o el AAN, es decir, a métodos de interpolación o contracción de neuronas (véanse las Figuras 6.30 y 6.83). Nótese que las clases de dimensión intrínseca uno quedan claramente reflejadas en las imágenes generadas por estos métodos para MAO bidimensionales. Naturalmente, otros métodos de proyección más simples (por ejemplo, lineales) pueden mostrar dichas estructuras, aunque en un espacio de observación diferente al del MAO. Con respecto a clases cuya dimensión intrínseca es mayor que la del MAO, su localización es bastante más compleja. Los mapas coloreados creados a partir de métodos de contracción pueden indicar *clusters* en el MAO donde

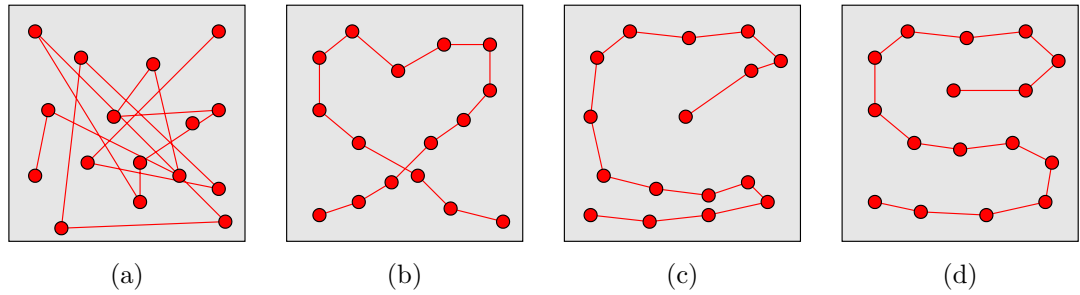


Figura 6.118: Tipos de discontinuidades de los MAO. En (a) un MAO 16-dimensional se ha adaptado a una distribución bidimensional cuadrada de forma totalmente desordenada. En (b) la adaptación aparece más ordenada pero aparece un “cruce” que puede dificultar el análisis visual. En (c) no hay cruces pero aparecen vectores de referencia lejanos en el mapa que representan datos muy similares, los cuales se hallan cerca de dichos vectores de referencia (algo que también ocurre en (b)). En (d) el MAO unidimensional se adapta “lo mejor posible” a la distribución, donde no aparecen cruces y las discontinuidades son debidas a datos relativamente lejos de los vectores de referencia (además, el error de cuantización es menor).

existen discontinuidades (generalmente asociadas a estructuras de dimensión elevada), si en ellos varias neuronas no-vecinas aparecen con colores similares (véase la Figura 6.87). No obstante, esta solución no siempre aporta resultados satisfactorios (en las Figuras 6.56 (d) y 6.57 (b) el color de las neuronas de cada *cluster* es prácticamente constante, lo cual no permite visualizar discontinuidades a pesar de que la dimensión intrínseca de los *clusters* es ocho). De esta manera, un primer objetivo consiste en diseñar un método de visualización capaz de revelar información sobre el grado de preservación de la topología de cada región del MAO, y sobre la dimensión intrínseca de los datos asociados a dichas regiones.

El propósito principal de los MAO es la visualización de relaciones no-lineales de datos multidimensionales [Koh01]. En la práctica, una de las situaciones más habituales consiste en trabajar con conjuntos de datos cuya dimensión intrínseca supera a la dimensión del MAO, ya que por razones de visualización ésta no puede ser mayor de tres. Por otro lado, la visualización de MAO tridimensionales es considerablemente más compleja que la relativa a MAO bidimensionales, dada la tecnología actual de visión tridimensional. Por esta razón, los analistas están “obligados” a trabajar con MAO bidimensionales para analizar distribuciones de dimensión elevada, donde la preservación de la topología rara vez se satisface en todas las regiones del MAO.

De esta manera, surge una nueva pregunta interesante: si el MAO se ha adaptado a la distribución “lo mejor posible” o por lo menos “adecuadamente” dentro

de sus posibilidades, cuando su dimensión es inferior a la dimensión intrínseca de los datos. La Figura 6.118 muestra esta idea mediante cuatro MAO unidimensionales (compuestos por 16 neuronas) adaptados a una distribución bidimensional cuadrada. En todos los casos no se preserva la topología, pero existen diferencias importantes de cara a analizar la distribución. En (a) el MAO aparece totalmente desordenado, lo cual impide visualizar la distribución. Los MAO en (b), (c) y (d) aparecen más ordenados ya que vectores de referencia cercanos en el espacio de los datos también se hallan cerca en el propio MAO. El cruce que aparece en (b) puede dificultar el análisis visual ya que neuronas alejadas en el mapa pueden representar datos muy similares cercanos a sus vectores de referencia, algo que también ocurre en (c). En (d) neuronas alejadas en el mapa pueden representar datos muy similares, pero éstos generalmente se hallan lejos de los vectores de referencia. De esta manera, dada la dimensión del MAO, la adaptación del cuarto MAO puede ser considerada “la mejor posible” (o al menos “adecuada”) para realizar un análisis visual. El segundo objetivo relativo a visualizar la calidad de la adaptación del MAO a una distribución consiste en diseñar un método capaz de detectar estas situaciones.

## 6.4.2. Solución propuesta

Para alcanzar los objetivos anteriores, este apartado propone la visualización explícita en el espacio de observación del MAO de grafos que unen neuronas cuyos vectores de referencia se hallan próximos en el espacio de los datos. Esta estrategia no ha sido utilizada previamente para analizar MAO de forma visual.

### 6.4.2.1. Grafo Hebbiano

Varias medidas de preservación de la topología están basadas en el principio de aprendizaje Hebbiano competitivo [Mar93] (el error topográfico [Kiv96] y la medida Hebbiana [Pol97]), donde los dos vectores de referencia más próximos a cada patrón de entrada son conectados para formar el grafo Hebbiano, y deben corresponder a neuronas vecinas en el mapa para que la topología sea preservada adecuadamente. Nótese que el principio está basado en la regla de Hebb [Heb49], donde se espera que exista una correlación entre las actividades de las neuronas asociadas a los dos vectores de referencia que mejor representan a un dato, por

lo que éstos son conectados. El grafo Hebbiano también es denominado “triangulación inducida de Delaunay” (véase [Mar93]). Si existe un dato cuyos vectores de referencia más próximos son  $w_i$  y  $w_j$ , entonces existe una región no vacía del diagrama de Voronoi de segundo orden asociada a  $w_i$  y  $w_j$ . Esto implica que la intersección de las regiones del diagrama de Voronoi de primer orden asociadas a  $w_i$  y  $w_j$  tampoco es vacía y la arista que los conecta forma parte de la triangulación de Delaunay de todos los vectores de referencia por definición. Por tanto, el grafo Hebbiano es un subgrafo de la triangulación de Delaunay, ya que pueden existir regiones del diagrama de Voronoi de segundo orden que no contengan datos.

Cuando el objetivo consiste en visualizar el grado de preservación de la topología de cada región del MAO una solución consiste en proyectar el grafo Hebbiano sobre su espacio de observación. El grafo proyectado está formado por aristas que conectan pares de neuronas cuyos vectores de referencia son los más cercanos a un dato de entrada. Por tanto, las aristas proyectadas que conectan neuronas vecinas en el mapa indican una buena preservación de la topología, mientras aristas más largas que conectan neuronas no-vecinas muestran discontinuidades en el MAO donde la topología no es preservada adecuadamente.

La visualización del grafo Hebbiano es una versión gráfica del error topográfico y la medida Hebbiana. El primero calcula la proporción de patrones de entrada con los cuales no se satisface el principio de aprendizaje Hebbiano competitivo, que es equivalente a observar (aproximadamente) el número de aristas que conectan neuronas no-vecinas. La medida Hebbiana compara el grafo Hebbiano con el grafo que conecta a todos los pares de neuronas vecinas del MAO. Desde un punto de vista gráfico, el proceso vuelve a ser similar a visualizar las aristas del grafo Hebbiano que conectan neuronas no-vecinas. La proyección del grafo Hebbiano permite visualizar los pares de neuronas que contribuyen a una buena o pobre preservación de la topología, por lo que es posible detectar zonas del MAO con diferentes grados de preservación de la topología, lo cual aporta una pista sobre la dimensión intrínseca de los datos asociados a dichas zonas.

El grafo Hebbiano contiene a lo sumo  $M(M - 1)/2$  aristas, donde  $M$  es el número de neuronas del MAO, aunque es necesario procesar  $N$  pares de vectores de referencia para calcularlo, donde  $N$  es el número de muestras de la distribución. Nótese que varios datos pueden originar la misma arista. Su complejidad computacional es  $\mathcal{O}(MN)$ . Aunque no es excesivamente elevada, si la distribución es de tamaño elevado el analista puede optar por proyectar otros grafos que

pueden ser calculados independientemente del número de muestras, como los dos presentados en los siguientes apartados.

Cuando el objetivo es visualizar la calidad de la adaptación del MAO cuando su dimensión es inferior a la dimensión intrínseca de los datos, el grafo Hebbiano produce imágenes con multitud de aristas conectando neuronas no-vecinas, que generalmente son difíciles de interpretar. El grafo Hebbiano no proporciona información sobre las distancias entre los pares de vectores de referencia que generan las discontinuidades, por lo que no puede determinar cruces del MAO o situaciones en las que neuronas lejanas en el mapa representan datos muy cercanos a sus respectivos vectores de referencia. No obstante, el subgrafo del grafo Hebbiano que conecta únicamente neuronas vecinas puede servir para determinar si el MAO se ha adaptado a los datos “lo mejor posible” (siempre que se haya comprobado previamente que el MAO no preserva adecuadamente la topología). Si este subgrafo conecta un porcentaje elevado de neuronas vecinas, es posible suponer que el MAO no puede adaptarse a la distribución de forma considerablemente más ordenada.

Otra posible solución al problema anterior consiste en ilustrar las aristas mediante niveles de gris, donde la intensidad luminosa de cada arista indique la “gravedad” de la posible discontinuidad de cara a la preservación de la topología. Por ejemplo, la intensidad luminosa puede depender del número de patrones asociados a una arista (par de vectores de referencia), o puede corresponder a una función creciente con respecto a la longitud de la arista y decreciente con respecto a las distancias entre el patrón y los dos vectores de referencia asociados a la arista. Sin embargo, aunque varias estrategias han sido probadas en esta dirección, las imágenes obtenidas no indican si el MAO se ha adaptado a los datos “lo mejor posible” con claridad. En este sentido, los otros dos grafos presentados en este apartado o el subgrafo Hebbiano que conecta neuronas vecinas presentan mejores resultados. En cualquier caso, el estudio de índices que indiquen la “gravedad” de las discontinuidades es propuesto como línea futura de esta tesis. Por ejemplo, al utilizar niveles de gris para representar el número de patrones asociados a cada arista del grafo Hebbiano, la imagen representa una versión “gráfica” más exacta de la medida Hebbiana.

Por último, el grafo Hebbiano puede revelar información sobre la estructura de *clusters*. Al añadir aristas asociadas a datos, las neuronas que no aparecen como vértices del grafo son interpolantes, y permiten detectar algunos bordes

entre *clusters*.

### 6.4.2.2. Proyección del árbol generador mínimo

El AGM también es un subgrafo de la triangulación de Delaunay, y es una simplificación del grafo Hebbiano, aunque no corresponde estrictamente a un subgrafo de éste. Contiene  $M - 1$  aristas, donde  $M$  es el número de neuronas del MAO (el grafo Hebbiano puede llegar a contener  $M(M - 1)/2$  aristas). Por esta razón no puede ilustrar todas las discontinuidades que aparecen en el MAO de acuerdo con el principio de aprendizaje Hebbiano competitivo, y su interpretación es diferente.

Sin embargo, ilustra algunas de las discontinuidades más importantes o posibles “graves”, ya que conecta neuronas cuyos vectores de referencia se encuentran próximos entre sí. De esta manera, el grafo Hebbiano ilustra mejor la preservación de la topología del MAO, mientras el AGM es capaz de dar una respuesta rápida y generalmente clara sobre la capacidad del MAO para adaptarse a los datos “lo mejor posible” (nótese que el subgrafo Hebbiano que conecta neuronas vecinas también aporta una visión aceptable en este sentido). Cuando las aristas del AGM conectan neuronas vecinas o muy próximas es posible suponer que el MAO se ha adaptado adecuadamente a la distribución. Si aparecen pocas aristas largas conectando neuronas lejanas en el mapa resulta probable que otro MAO pueda adaptarse mejor a los datos. Por último, si aparecen muchas aristas largas el AGM no aporta información fiable sobre la capacidad del MAO para adaptarse a los datos “lo mejor posible”.

Las aristas del AGM también pueden ser mostradas mediante niveles de gris. Por ejemplo, en función de la distancia entre los vectores de referencia asociados a cada arista u otras estrategias como el número de patrones asociados al par de vectores de referencia. Sin embargo, al igual que el resto de grafos presentados en este apartado, el estudio de éstos grafos coloreados es propuesto como línea futura de la tesis, ya que los resultados obtenidos mediante estas estrategias no aportan información útil excepto cuando el tamaño del MAO es muy pequeño (por ejemplo, menos de 100 neuronas).

Por último, su complejidad computacional es  $\mathcal{O}(M^2)$  (por ejemplo, mediante el popular método de Prim [Pri57]), donde  $M$  es el número de neuronas del MAO. Nótese que ésta puede ser reducida a  $\mathcal{O}(M \lg M)$  cuando los vectores de referencia están definidos en un plano (el espacio de los datos sería bidimensional),

construyendo primero la triangulación de Delaunay.

#### 6.4.2.3. Grafo de los $K$ -vecinos más cercanos

El grafo presentado en este apartado es similar al AGM en el sentido de no representar aristas entre pares de neuronas de acuerdo con el principio de aprendizaje Hebbiano competitivo. En cambio, trata de ilustrar otras conexiones importantes o posibles “graves” que deben ser cortas ya que unen vectores de referencia próximos entre sí en el espacio de los datos. En concreto el grafo conecta cada neurona  $\eta_i$  con el conjunto de neuronas asociadas a los  $K$  primeros vectores de referencia más cercanos (vecinos en el espacio de los datos) al de  $\eta_i$ . Este grafo puede ser utilizado para visualizar la capacidad del MAO para adaptarse a los datos “lo mejor posible”, y también para analizar la preservación de la topología general.

Cuando  $K = 1$  el grafo contiene  $M$  aristas como máximo (donde  $M$  es el número de neuronas del MAO), es similar al AGM y representa las principales aristas que deben ser cortas y conectar neuronas vecinas. Con  $K$  relativamente bajo las aristas deben seguir siendo cortas y los grafos aportan una buena visión sobre la calidad de la adaptación del MAO a la distribución cuando ésta presenta una dimensión intrínseca más elevada que la dimensión del MAO. A medida que  $K$  aumenta las aristas empiezan a conectar neuronas no-vecinas y el grafo empieza a parecerse al propio grafo Hebbiano, por lo que puede ser utilizado como medida aproximada de preservación de la topología general, pero no para evaluar si el MAO se ha adaptado a los datos “lo mejor posible”. Si el MAO tuviera una topología rectangular y preservara perfectamente la topología, cada neurona del grafo Hebbiano debería conectar exactamente ocho vecinos en el interior del mapa, cinco en los laterales, y tres en las esquinas. En ese caso  $K$  podría ser elevado hasta a ocho (modificando el algoritmo para reducir el número de aristas asociadas a las neuronas de los bordes del mapa) sin que aparecieran aristas conectando neuronas no-vecinas. No obstante, si la dimensión intrínseca de los datos es elevada esta situación no ocurre (cada neurona no va aparecer conectada exactamente a todas sus vecinas) y sólo es posible evaluar si el MAO se ha adaptado a los datos “lo mejor posible” con un valor de  $K$  relativamente bajo.

Su complejidad computacional es  $\mathcal{O}(M^2)$ , donde  $M$  es el número de neuronas del MAO. Puede ser utilizado como sustituto del AGM (para  $K = 1$  o  $K = 2$ )



ya que posee la misma complejidad computacional.

### 6.4.3. Resultados experimentales

Este apartado describe varios experimentos con el objetivo de ilustrar la información que aportan los grafos propuestos. Los MAO utilizados son de tamaño medio o pequeño ya que generalmente resulta difícil interpretar las proyecciones de grafos cuando éstos contienen un número elevado de aristas. Como el tamaño de las distribuciones tampoco es excesivamente elevado, las imágenes son generadas prácticamente en tiempo real con el sistema utilizado para llevar a cabo los experimentos de esta tesis (ordenador personal con procesador Intel® Pentium® 4 a 1.7GHz con 256MB de memoria RAM).

#### 6.4.3.1. Ocho *clusters* disjuntos con forma hipercúbica

La Figura 6.119 muestra el primer ejemplo de la proyección de grafos, el cual está basado en el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica utilizado a lo largo de la tesis. Ilustra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y los grafos de los  $K$ -vecinos más cercanos para  $K = 1$ , 2 y 8, en (d), (e) y (f), respectivamente.

La proyección del grafo Hebbiano muestra las 20000 aristas correspondientes a cada dato de la distribución. Aunque muchas de éstas aparecen repetidas (varios datos tienen el mismo par de vectores de referencia más cercanos) la imagen resulta ruidosa, en el sentido de que no permite visualizar cada arista con claridad. Esto es debido al elevado número de aristas y a su longitud relativamente elevada, donde muchas de ellas conectan neuronas no-vecinas. A diferencia del resto de grafos proyectados, los grafos “Hebbianos” en (a) y (b) sólo conectan neuronas asociadas a datos de la distribución. Es decir, no conectan neuronas interpolantes, lo cual permite detectarlas con claridad, y por tanto los bordes entre los *clusters*. Nótese la similitud entre la proyección de grafo Hebbiano y las imágenes obtenidas con la mayoría de métodos de visualización de MAO (véanse las Figuras 5.2, 5.4, 5.8 (a), 5.9, 5.12, 5.17 (a), 5.20, 5.22, 5.23 (a), 5.26, 6.46, 6.56 (d) y 6.57 (b)). En cuanto a la preservación de la topología, queda claro que un MAO bidimensional no puede adaptarse ordenadamente a los *clusters* (cuya dimensión intrínseca es ocho), por lo que aparecen numerosas aristas que conectan neuronas no-vecinas.

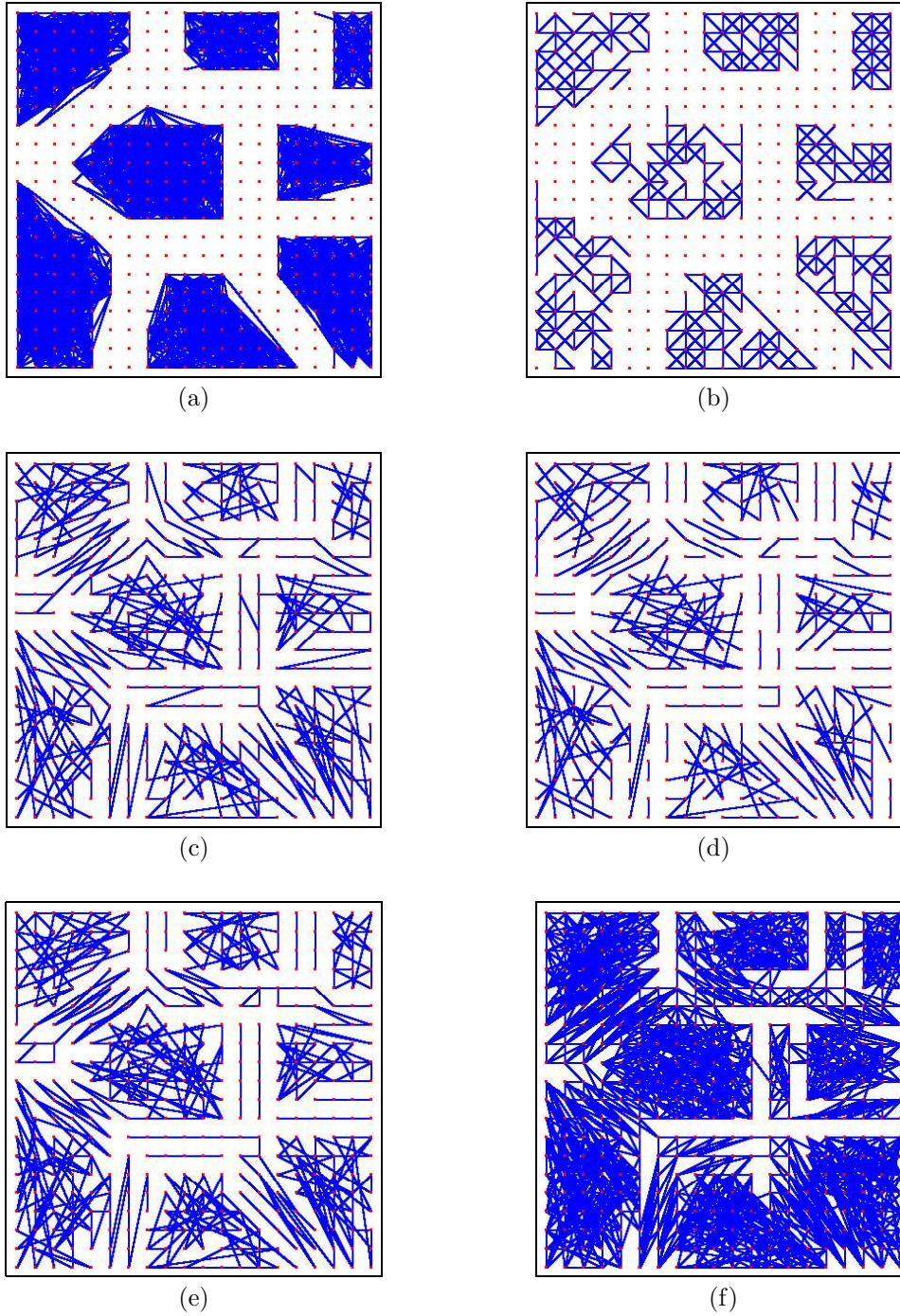


Figura 6.119: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por ocho *clusters* disjuntos con forma hipercúbica. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d),  $K = 2$  (e) y  $K = 8$  (f).

No obstante, el subgrafo Hebbiano en (b) contiene un elevado número de aristas conectando neuronas vecinas en cada *cluster*, lo cual indica que la adaptación del MAO es aceptable, teniendo en cuenta que la dimensión intrínseca de la distribución es muy superior a la dimensión del MAO.

Aunque la interpretación del AGM es diferente que la del grafo Hebbiano, su proyección también permite detectar la pobre preservación de la topología ya que aparece un elevado número de aristas que conectan neuronas no-vecinas. Generalmente éstas son numerosas cuando el MAO está totalmente desordenado o cuando la dimensión intrínseca de los datos es muy elevada, como ocurre en este ejemplo. Por otro lado, el AGM sólo indica que el MAO ha sido adaptado a la distribución “lo mejor posible” cuando éste contiene una mayoría de aristas “cortas”, conectando neuronas cercanas en el mapa. En otras palabras, cuando aparecen muchas aristas “largas” el AGM no es capaz de revelar si el MAO está totalmente desordenado o si ha sido capaz de adaptarse “lo mejor posible” a los datos. En este caso el analista puede recurrir al subgrafo Hebbiano que conecta neuronas vecinas para medir la calidad de la adaptación del MAO a una distribución de dimensión intrínseca elevada. Por otro lado, es posible observar como las zonas donde el MAO aparece más desordenado corresponden a los *clusters*, mientras el grado de desorden es menor en los bordes entre ellos.

Con respecto a los grafos de los  $K$ -vecinos más cercanos para  $K = 1$  y  $K = 2$ , éstos son muy parecidos al AGM. No pueden ilustrar si el MAO ha sido adaptado a la distribución “lo mejor posible” por la misma razón que el AGM. Las discontinuidades son muy numerosas por lo que sólo pueden indicar un elevado desorden en el MAO, pero no la calidad de la adaptación. Por otro lado, cuando  $K = 8$  el grafo es parecido al grafo Hebbiano salvo en las conexiones asociadas a las neuronas interpolantes.

### 6.4.3.2. Cuatro *clusters* disjuntos de diferente tamaño e igual forma

El siguiente experimento es similar al anterior ya que está basado en la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma, donde la dimensión intrínseca de los *clusters* es ocho. La Figura 6.120 muestra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución, véase la Figura 6.28.

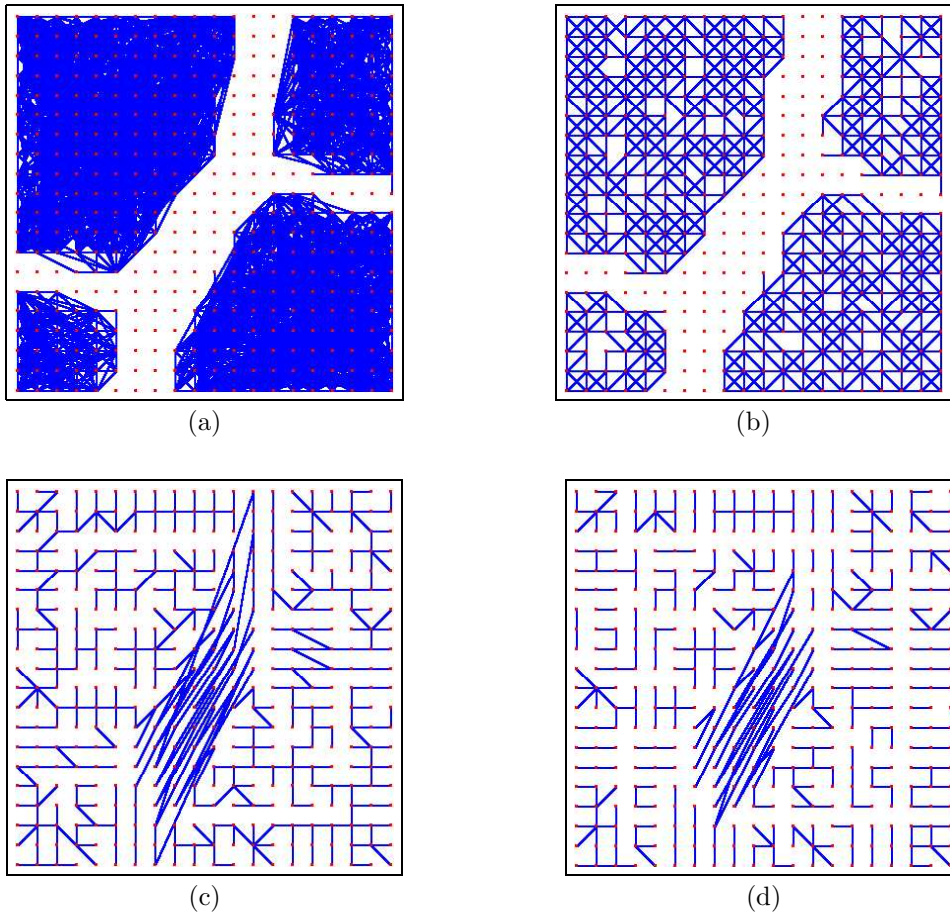


Figura 6.120: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por cuatro *clusters* disjuntos de diferente tamaño e igual forma. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

Dada la elevada dimensión intrínseca de los *clusters*, el grafo Hebbiano muestra numerosas aristas que conectan neuronas no-vecinas, indicando que el MAO no puede adaptarse a ellos de forma ordenada. Aunque la imagen resulta ruidosa, es posible discriminar los cuatro *clusters* con claridad ya que las neuronas interpolantes asociadas a los bordes entre ellos no pueden ser vértices del grafo Hebbiano. De esta manera, la imagen resultante aporta información similar a otros métodos de visualización (véanse las Figuras 6.45, 6.58 y 6.59). Por otro lado, el subgrafo Hebbiano que conecta únicamente neuronas vecinas contiene un elevado número de aristas en los *clusters*. Cada neurona aparece conectada a la mayoría de sus vecinas, lo cual indica que el MAO se ha adaptado “lo mejor posible” en los *clusters*, teniendo en cuenta la dimensión intrínseca elevada de éstos. La proyección del AGM y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$ , que resultan muy similares, también indican que el MAO se ha adaptado a los *clusters* adecuadamente ya que no aparecen aristas conectando neuronas no-vecinas en ellos. Éstas imágenes también muestran el desorden del MAO en su centro, asociado al mayor borde entre los *clusters*. Aunque esta información no puede ser vista mediante el grafo Hebbiano, en este caso no resulta importante de cara al análisis de la distribución.

##### 6.4.3.3. Círculo y anillo

La dimensión intrínseca de los datos puede coincidir con la dimensión del MAO, en cuyo caso éste debe adaptarse a la distribución de forma ordenada. La Figura 6.121 ilustra una serie de grafos para un MAO  $(20 \times 20)$ -dimensional (bidimensional) adaptado a la distribución formada por un círculo y un anillo, cuya dimensión intrínseca es claramente dos. La figura muestra el grafo Hebbiano (a), el AGM (b), el grafo de los  $K$ -vecinos más cercanos para  $K = 4$  (c), y  $K = 8$  (d).

En el grafo Hebbiano la mayoría de las aristas conectan neuronas vecinas, lo cual indica una muy buena preservación de la topología. Al igual que en los ejemplos anteriores, las neuronas interpolantes que forman parte de los bordes entre las dos clases son claramente visibles al no estar conectadas al grafo. De esta manera, la proyección del grafo Hebbiano permite discriminar dos clases en los datos, donde la imagen resultante está relacionada con las generadas mediante otros métodos de visualización (véanse las Figuras 6.44, 6.84 y 6.85). En general, el AGM tampoco muestra aristas conectando neuronas no-vecinas, por lo que



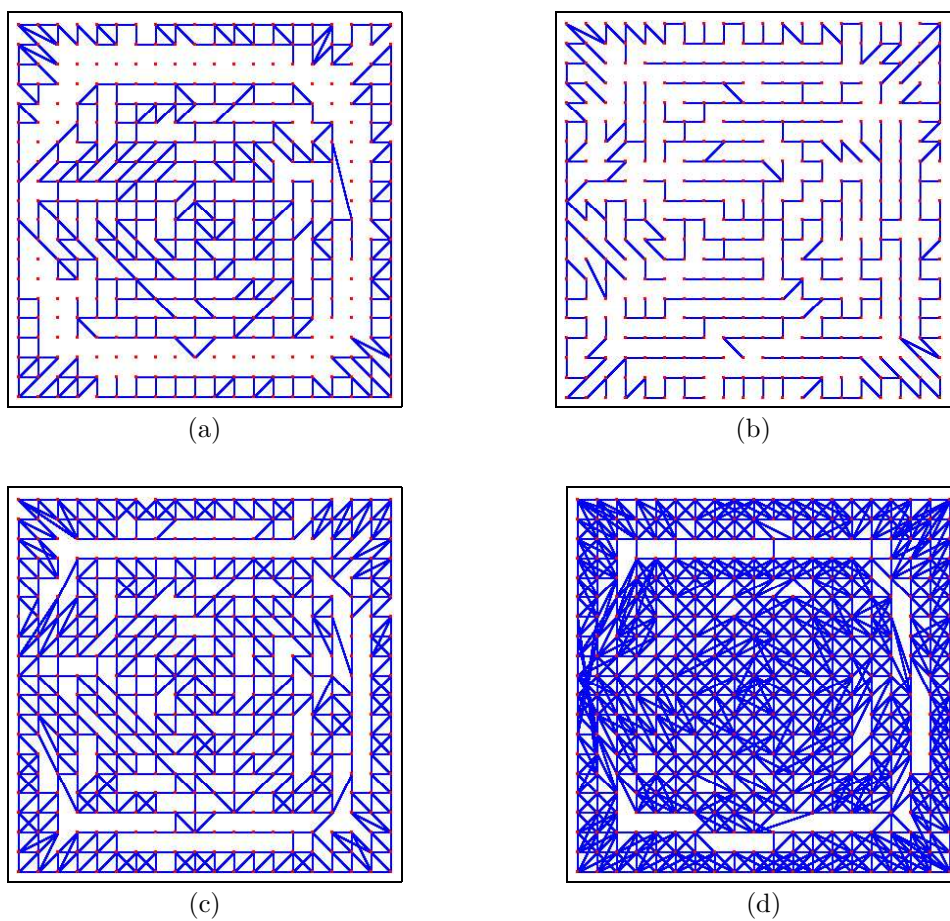


Figura 6.121: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por un círculo y un anillo. Grafo Hebbiano (a), AGM (b), y grafo de los  $K$ -vecinos más cercanos para  $K = 4$  (c) y  $K = 8$  (d).

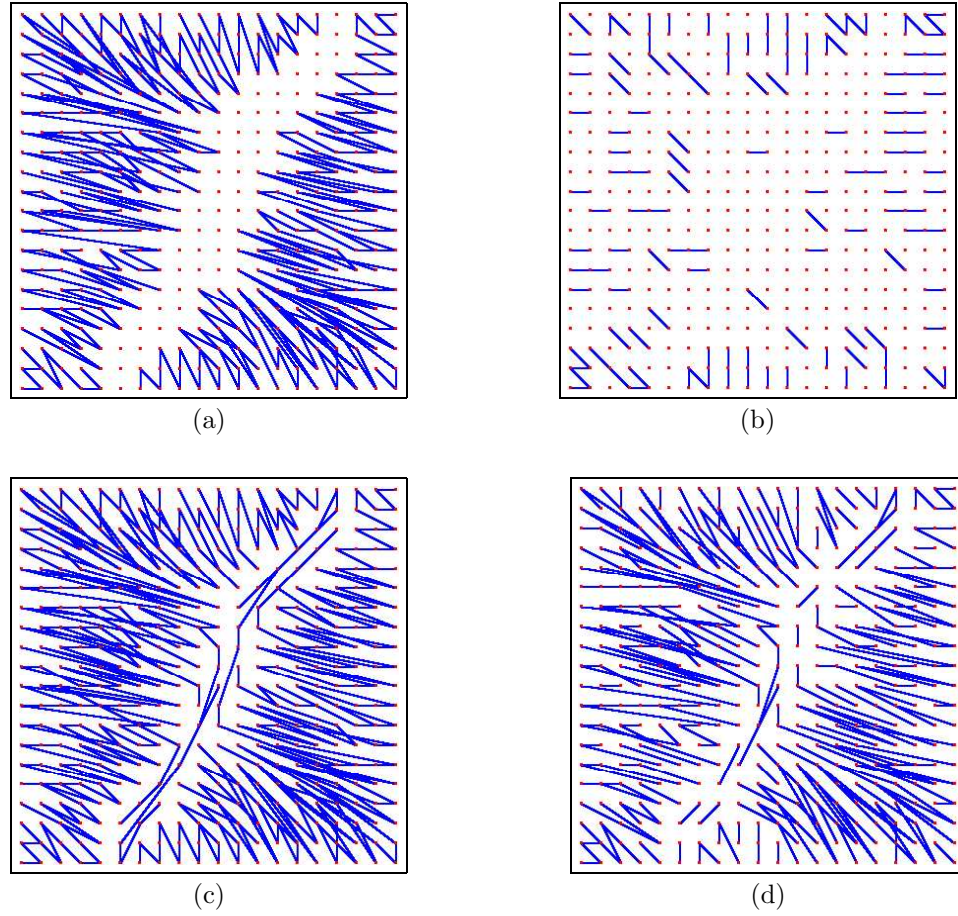


Figura 6.122: Grafos proyectados para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos segmentos. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

confirma la buena preservación de la topología del MAO. Las pocas conexiones entre neuronas no-vecinas no son relevantes al seguir siendo cortas. Con respecto al grafo de los  $K$ -vecinos más cercanos, cuando  $K$  es bajo las imágenes muestran la misma información que el AGM. Es necesario elevar  $K$  (por ejemplo, hasta cuatro) para visualizar aristas que conecten neuronas no-vecinas. Sin embargo, sus longitudes siguen siendo pequeñas, lo cual es otro indicativo de la buena preservación de la topología. Con  $K = 8$  las longitudes de las aristas todavía son pequeñas y destacan los claros bordes entre las dos clases.

#### 6.4.3.4. Dos segmentos

En este experimento la distribución de datos utilizada corresponde a la formada por dos segmentos, cuya dimensión intrínseca es uno, que es menor que la dimensión del MAO. En este caso se espera que aparezcan aristas largas conectando neuronas no vecinas, pero la interpretación de las imágenes es ligeramente diferente ya que no aparece ninguna de las situaciones descritas en la Sección 6.4.1 (véase la Figura 6.118), donde la dimensión del MAO es menor que la intrínseca de los datos. La Figura 6.122 muestra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución, véase la Figura 6.77.

El grafo Hebbiano discrimina las dos clases correspondientes a los segmentos al mostrar un claro borde de neuronas interpolantes en el centro del mapa, por lo que revela información similar a la obtenida mediante otros métodos de visualización (véanse las Figuras 6.76, 6.78 y 6.79). También muestra numerosas aristas que conectan neuronas no-vecinas, lo cual indica una pobre preservación de la topología. Por otro lado, el subgrafo Hebbiano que conecta neuronas vecinas contiene muy pocas aristas. En principio, esto podría indicar que es posible entrenar el MAO de forma que se adapte más ordenadamente a los datos. Sin embargo, las aristas del grafo Hebbiano muestran claras estructuras en zigzag, características de los MAO bidimensionales al adaptarse a datos de dimensión intrínseca uno. Cuando la dimensión intrínseca de los datos es superior a la dimensión del MAO éste puede doblarse para adaptarse de manera relativamente ordenada a los datos (véase la Figura 6.118). Sin embargo, cuando es inferior resulta imposible adaptar los vectores de referencia a los datos preservando el orden local de las neuronas en el mapa, por lo que aparecen dichas estructuras en zigzag. De esta manera, el propio grafo Hebbiano indica mediante estas estructuras que la dimensión intrínseca de los datos es uno y que es posible suponer que el MAO se ha adaptado “lo mejor posible” a la distribución. Nótese que en estos casos el MAO no se adapta de manera aleatoria, sino que presenta cierto orden y aparece una estructura en zigzag, reflejando el intento del MAO por adaptarse a los datos de dimensión intrínseca uno. Por otro lado, el AGM y el grafo de los  $K$ -vecinos más cercanos con  $K = 1$  son muy similares al grafo Hebbiano. Éstos revelan aristas largas formando estructuras en zigzag por lo que también indican una pobre preservación de la topología, la dimensión intrínseca de los datos (uno)



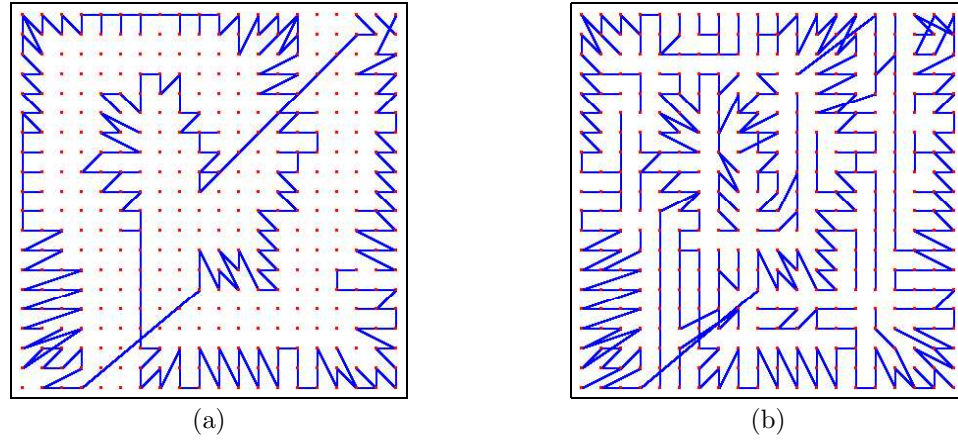


Figura 6.123: Grafos proyectados para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados. Grafo Hebbiano (a) y AGM (b).

y una calidad aceptable del proceso de adaptación del MAO.

#### 6.4.3.5. Dos eslabones encadenados

La Figura 6.123 muestra el grafo Hebbiano (a) y el AGM (b) para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos eslabones encadenados, véase la Figura 5.6 (a). El grafo Hebbiano vuelve a mostrar una imagen similar a las generadas con otros métodos de visualización (véanse las Figuras 5.6, 5.17 (b), 5.23 (b), 5.40 (a), 5.44 (a), 6.30, 6.47 y 6.83). Éste revela la existencia de dos clases al mostrar las neuronas interpolantes, la dimensión intrínseca de los datos (uno, debido a la organización de las aristas en zigzag) y, a diferencia de los métodos tradicionales, es capaz de ilustrar la estructura circular de las clases al conectar mediante dos aristas largas las zonas del mapa donde aparecen las discontinuidades más notables. En este caso, el MAO ha sido adaptado a los datos adecuadamente a pesar de dichas discontinuidades, necesarias para indicar que los dos anillos están encadenados. Sin embargo, si no se conoce la estructura de los datos *a priori* no es posible determinar si el MAO ha sido adaptado a los datos “lo mejor posible” únicamente a partir de los grafos proyectados. Las aristas largas del grafo Hebbiano hacen sospechar que pueda existir otro MAO que pueda adaptarse a los datos de forma más ordenada. Por otro lado, la proyección del AGM genera una imagen mucho más ruidosa que la generada mediante el grafo Hebbiano al conectar neuronas interpolantes y no aporta información útil sobre la distribución. La proyección del grafo de los  $K$ -vecinos más cercanos para

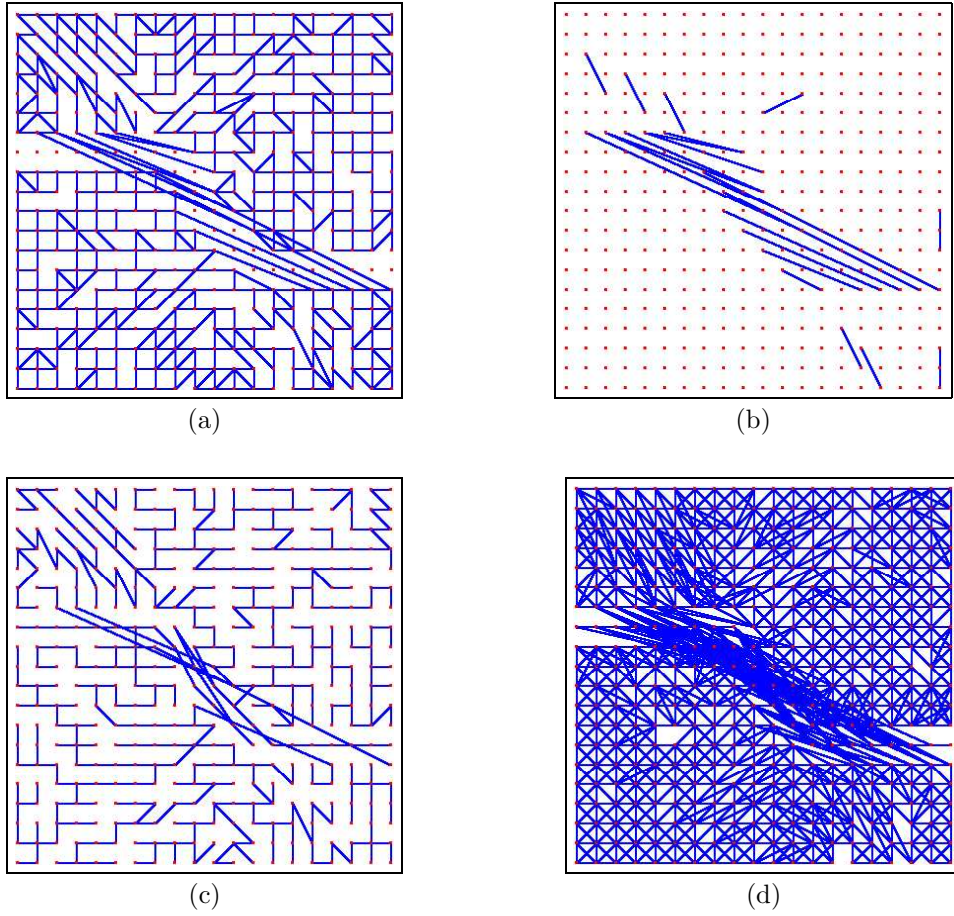


Figura 6.124: Grafos proyectados para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos superficies planas paralelas. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas no-vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 8$  (d).

valores bajos de  $K$  es similar a la del AGM por lo que no es mostrada en el ejemplo.

#### 6.4.3.6. Dos superficies planas paralelas

El siguiente experimento está basado en un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución formada por dos superficies planas paralelas, véase la Figura 6.26. La Figura 6.124 muestra el correspondiente grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas no-vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 8$  (d).

Las imágenes generadas con métodos de visualización tradicionales logran discriminar ambas clases pero no consiguen determinar la dimensión intrínseca de

los datos. De esta manera, éstas son parecidas a las obtenidas con otras distribuciones compuestas por dos clases disjuntas (que contienen aproximadamente el mismo número de patrones de entrada), como ocurre con la distribución formada por dos segmentos. No obstante, la proyección del grafo Hebbiano no sólo está relacionada con otras imágenes generadas por métodos de visualización (véanse las Figuras 6.89, 6.90 y 6.91), sino que permite detectar la dimensión intrínseca de las dos clases. El grafo Hebbiano consigue discriminar ambas clases al revelar un borde de neuronas interpolantes en el centro del mapa. Las aristas que conectan neuronas no-vecinas en dicho borde indican que el MAO se ha doblado en el entrenamiento, por lo que parece factible entrenar otro MAO que quede ajustado mejor a la distribución. Por otro lado, prácticamente la totalidad de las conexiones asociadas a las dos superficies corresponden a neuronas vecinas (véase la imagen (b)), lo cual indica que su dimensión intrínseca es dos. El AGM revela la misma información que el grafo Hebbiano, aunque no es capaz de revelar el borde entre las dos superficies. Por último, el grafo de los  $K$ -vecinos más cercanos para  $K = 8$  indica un muy buen ordenamiento del MAO en las clases, donde numerosas neuronas aparecen conectadas únicamente a sus neuronas vecinas. Esto indica claramente que los datos se hallan sobre hipersuperficies de dimensión dos.

##### 6.4.3.7. Circunferencia, segmento, cuadrado y ortoedro

Este apartado presenta uno de los ejemplos más interesantes de la proyección de grafos. La distribución corresponde a la formada por una circunferencia, un segmento, un cuadrado y un ortoedro, por lo que consta de varias clases con diferente dimensión intrínseca. La Figura 6.125 ilustra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d), para un MAO  $(20 \times 20)$ -dimensional adaptado a la distribución, véase la Figura 5.13.

La proyección del grafo Hebbiano vuelve a estar relacionada con las imágenes producidas mediante otros métodos de visualización (véanse las Figuras 5.13, 5.40 (b), 5.44 (b), 6.48 y 6.95), y muestra las cuatro clases que componen la distribución con claridad al mostrar el grado de preservación de la topología en cada zona del mapa. La circunferencia aparece mediante una estructura circular en zigzag (por lo que su dimensión intrínseca es uno) en la esquina superior-derecha. El segmento aparece algo más abajo, representado por numerosas aristas que conectan neuronas no-vecinas formando otra estructura en zigzag. Queda claro

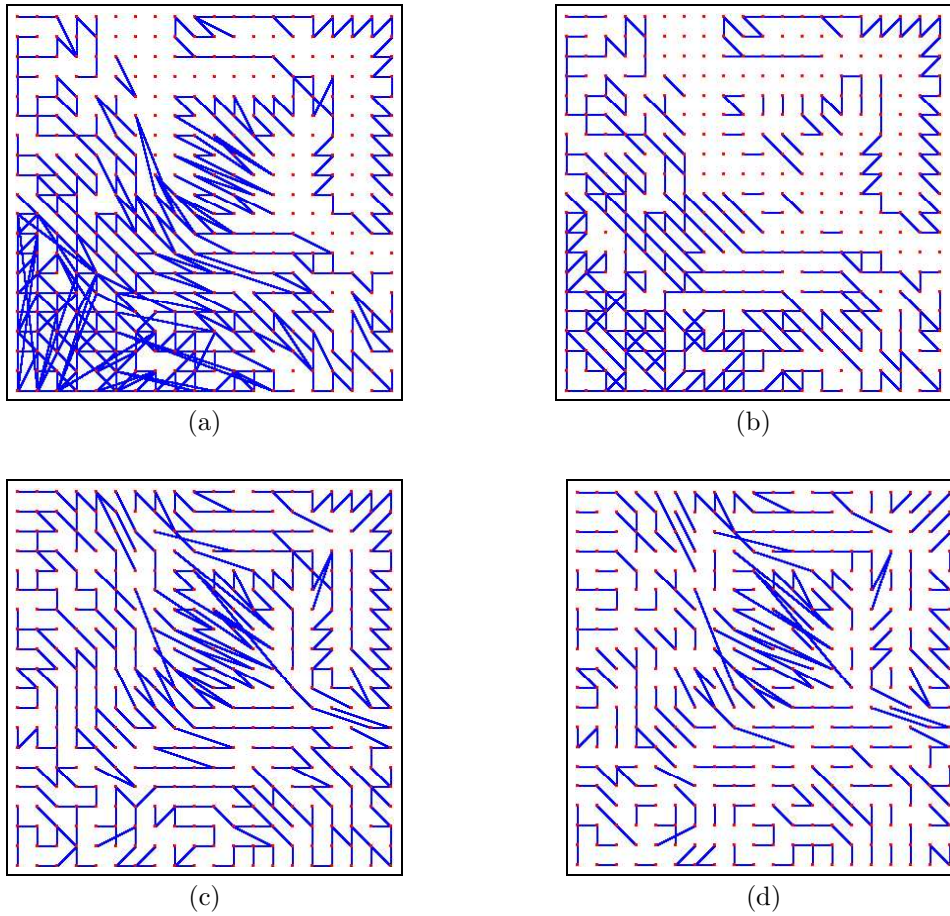


Figura 6.125: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una circunferencia, un segmento, un cuadrado y un ortoedro. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

en el ejemplo que el MAO se adapta mejor a la circunferencia que al segmento. La clase correspondiente al ortoedro también es claramente visible en la esquina inferior-izquierda, ya que aparece una zona donde las aristas conectan numerosas neuronas no-vecinas de forma desordenada, lo cual revela una clase de datos cuya dimensión intrínseca es superior a dos. Por último, el cuadrado se encuentra entre el segmento y el ortoedro, donde el MAO preserva adecuadamente la topología al aparecer menos discontinuidades.

A la hora de analizar la calidad de la adaptación del MAO al ortoedro, el grafo Hebbiano que conecta neuronas vecinas muestra un porcentaje elevado de conexiones entre las neuronas del ortoedro, lo cual indica que dicha adaptación es adecuada. Esto queda corroborado mediante las proyecciones del AGM y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$ , las cuales no muestran discontinuidades en la zona del mapa asociada al ortoedro. Con respecto al resto del MAO, las imágenes también muestran una adaptación adecuada, teniendo en cuenta que las aristas que conectan neuronas no-vecinas corresponden al segmento (datos de dimensión intrínseca uno), al que no puede adaptarse ordenadamente un MAO bidimensional.

##### 6.4.3.8. Esfera maciza dentro de una esfera hueca

La Figura 6.126 muestra otro ejemplo basado en el MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca, véase la Figura 6.31. Ilustra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

En este caso la proyección del grafo Hebbiano muestra una imagen ruidosa, debido al elevado número de aristas que conectan neuronas no-vecinas asociadas a la esfera maciza. Las numerosas discontinuidades indican una elevada dimensión intrínseca de los datos asociados a dicha clase. Por otro lado, la adaptación del MAO a la esfera hueca, cuya dimensión intrínseca puede ser considerada cercana a dos, es considerablemente mejor. En las regiones del mapa asociadas a esta clase las aristas del grafo Hebbiano conectan principalmente neuronas vecinas. De esta manera, el grafo Hebbiano aporta una visión adecuada de la preservación de la topología en cada región del MAO.

La proyección del grafo Hebbiano que conecta neuronas vecinas está relacionada con otras imágenes generadas con varios métodos de visualización, ya que



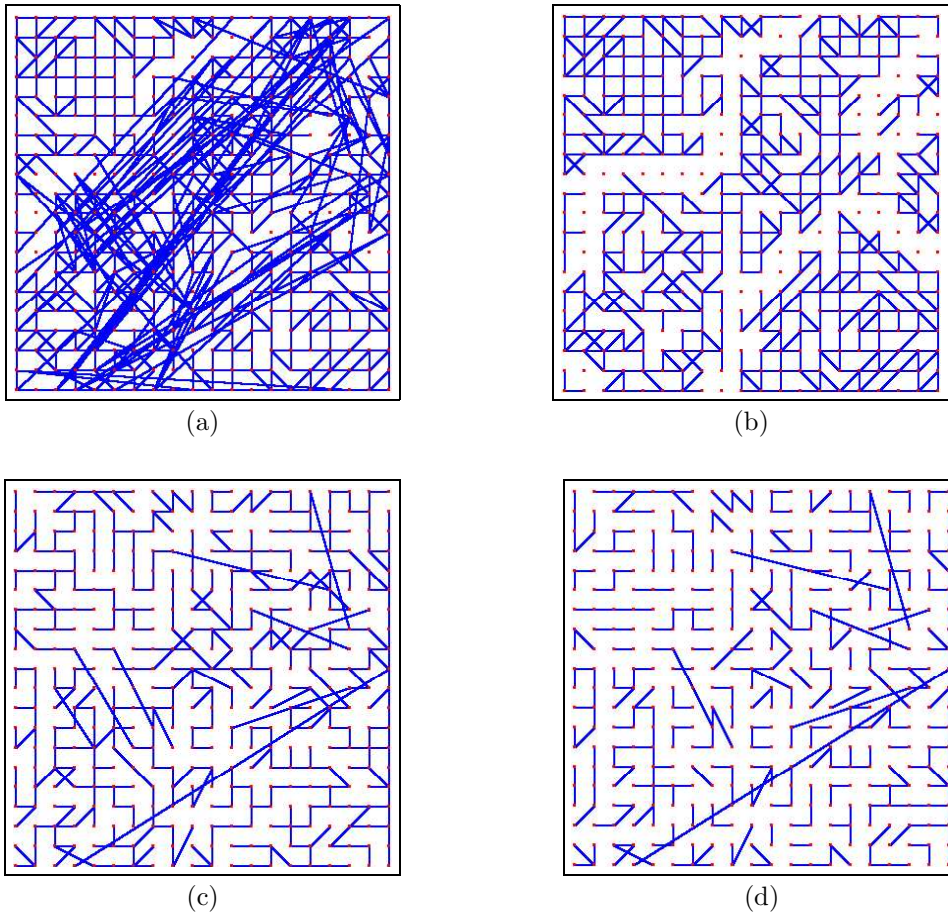


Figura 6.126: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por una esfera maciza dentro de otra hueca. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

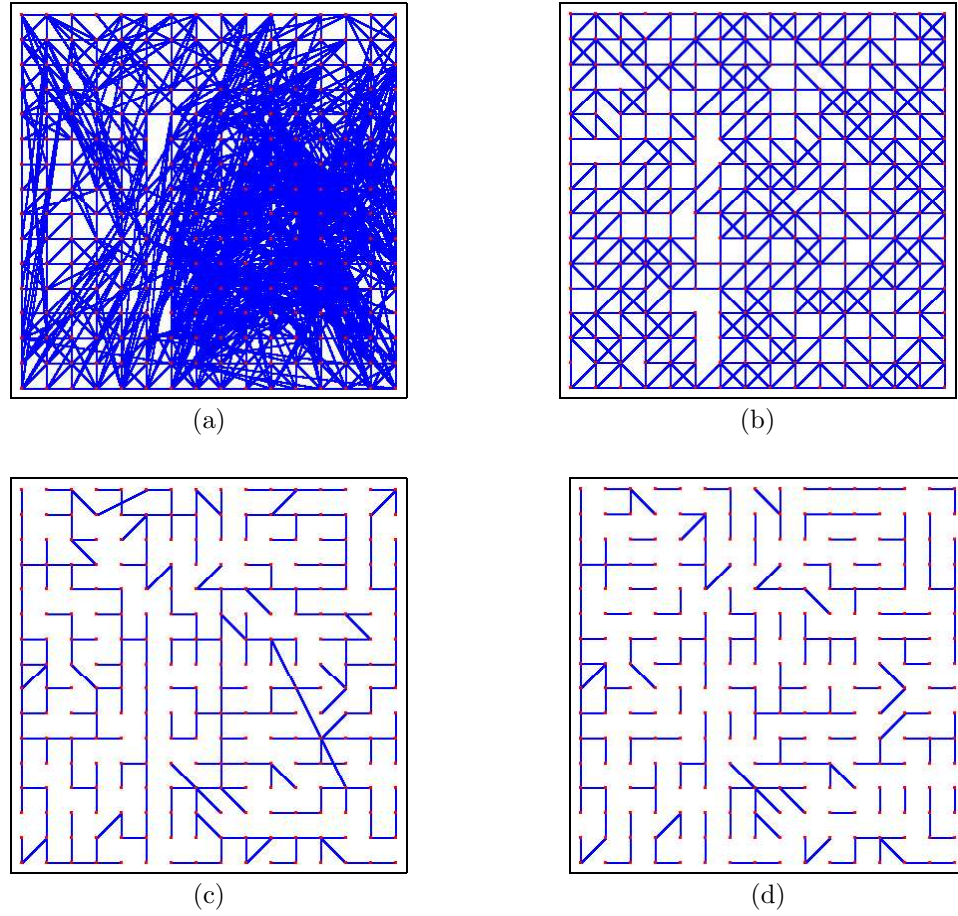


Figura 6.127: Grafos proyectados para un MAO ( $20 \times 20$ )-dimensional adaptado a la distribución formada por datos reales de habla. Grafo Hebbiano (a), subgrafo Hebbiano conectando neuronas vecinas (b), AGM (c), y grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

permite detectar bordes entre clases (véanse las Figuras 6.86, 6.87 y 6.88). Sin embargo, de cara a observar si el MAO ha sido adaptado a la esfera maciza “lo mejor posible”, no aporta información concluyente ya que la media de conexiones a neuronas vecinas no es elevada. En cambio, el AGM y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$ , que son muy similares, muestran un número relativamente bajo de aristas conectando neuronas no-vecinas. Esto indica que puede ser posible entrenar un MAO de manera que quede mejor adaptado a la distribución.

#### 6.4.3.9. Datos reales de habla

La Figura 6.127 muestra varias proyecciones de grafos para un MAO ( $16 \times 16$ )-dimensional adaptado a la distribución formada por datos reales de habla de la

palabra “*right*”, véase la Figura 6.49 (d). Ilustra el grafo Hebbiano (a), el subgrafo Hebbiano que sólo conecta neuronas vecinas (b), el AGM (c), y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  (d).

El grafo Hebbiano produce una imagen con multitud de aristas conectando neuronas no-vecinas, lo cual indica que la dimensión intrínseca de los datos es superior a dos. El elevado número de discontinuidades genera una imagen ruidosa y difícil de interpretar, donde su relación con otras imágenes obtenidas a partir de métodos de visualización no es tan evidente como en otros ejemplos (véanse las Figuras 6.49, 6.96 y 6.100).

No obstante, la adaptación del MAO a la distribución puede ser considerada adecuada o “la mejor posible” dada la información que aportan el resto de grafos proyectados. El grafo Hebbiano que sólo conecta neuronas vecinas contiene un elevado número de aristas, donde cada neurona suele estar conectada con la mayoría de sus vecinas. El MAO apenas muestra discontinuidades y las pocas que aparecen conectan neuronas relativamente próximas entre sí. Por último, el AGM y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  tampoco muestran discontinuidades serias del MAO.

La Figura 6.101 ilustra un ejemplo con datos de habla de la palabra inglesa “*up*”, donde es posible detectar una discontinuidad importante en el AGM. Las neuronas asociadas a la arista más larga corresponden al mismo tipo de datos, por lo que aparece un *cluster* “repetido” (de silencio). Esto indica que el MAO empleado no se ha adaptado a los datos “lo mejor posible”, y el analista puede entrenar otro MAO para eliminar uno de los *clusters* repetidos. En este ejemplo, el grafo Hebbiano muestra numerosas discontinuidades, el subgrafo Hebbiano está compuesto por un elevado número de aristas y el grafo de los  $K$ -vecinos más cercanos para  $K = 1$  no muestra discontinuidades. De esta manera, el AGM es el que aporta la información necesaria para permitir ver la discontinuidad importante.

#### 6.4.3.10. Proyección de grafos con MAO pequeños

La proyección de grafos puede ser utilizada como alternativa a los métodos tradicionales de visualización cuando el tamaño del MAO es pequeño. La Figura 6.128 muestra varios grafos que ilustran información acerca de la estructura de *clusters* de la distribución formada por cuatro *clusters* de igual tamaño y diferente forma, a la que se ha adaptado un MAO  $(5 \times 5)$ -dimensional (véase la Figura 6.19 (a)). Los grafos coloreados, que resultan difíciles de analizar cuando el tamaño



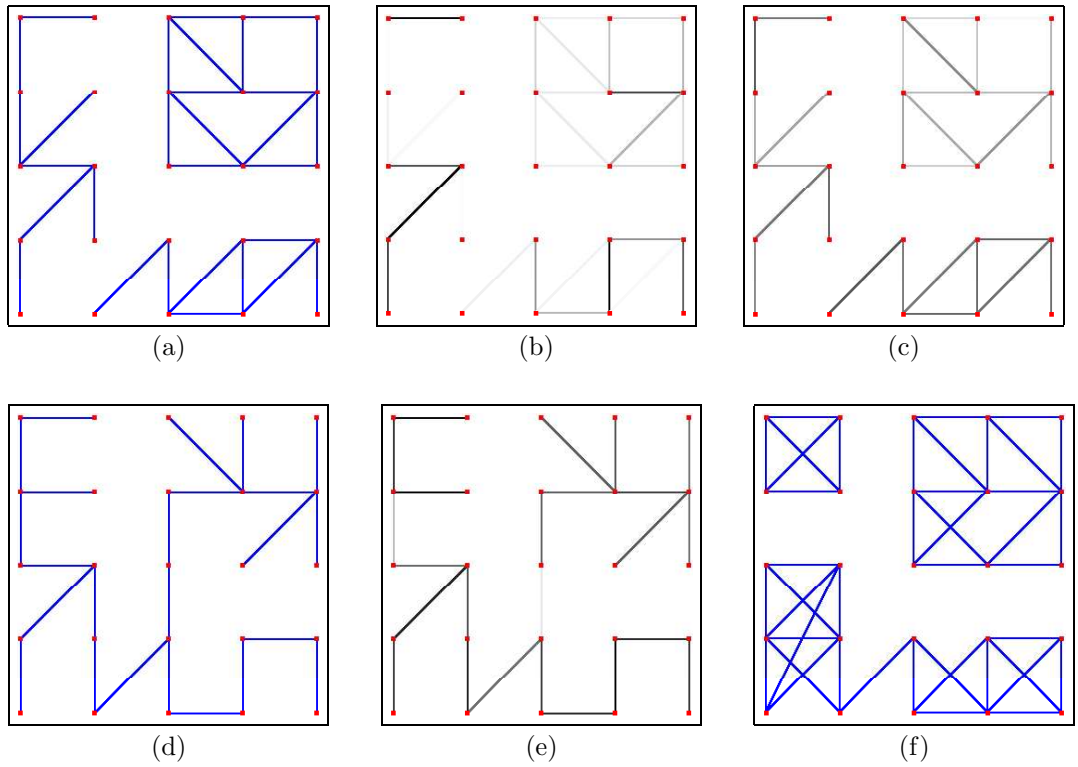


Figura 6.128: Grafos proyectados para un MAO  $(5 \times 5)$ -dimensional adaptado a la distribución formada por cuatro *clusters* de igual tamaño y diferente forma. Grafo Hebbiano (a), grafo Hebbiano coloreado según el número de patrones asociados a cada arista (b), grafo Hebbiano coloreado según la longitud de las aristas en el espacio de los datos (c), AGM (d), AGM coloreado según la longitud de las aristas en el espacio de los datos (e), y grafo de los  $K$ -vecinos más cercanos para  $K = 3$  (f).

del MAO es medio o elevado, ayudan a discriminar las clases que aparecen en los datos. En la imagen (b) las aristas reciben un color más oscuro a medida que representan más datos, por lo que la imagen está relacionada con el histograma de datos. En (c) y (e) las aristas son más claras a medida que aumenta la distancia entre los dos vectores de referencia asociados a las aristas, por lo que están relacionadas con matrices de distancias (véase la Figura 6.40).

#### 6.4.4. Discusión

El propósito de la estrategia de proyección de grafos es aportar una medida visual e intuitiva de preservación de la topología, por lo que las visualizaciones no pueden ser comparadas con las generadas por el resto de métodos descritos en esta tesis, aunque en algunos casos existe una relación entre ellas. Por otro lado, el autor de esta tesis no ha encontrado ningún otro método de proyección de grafos en la literatura para análisis visual de MAO. El método puede ser utilizado con MAO de cualquier tamaño, aunque proporciona mejores resultados con los de tamaño pequeño o medio (menos de mil neuronas aproximadamente) ya que en MAO grandes las conexiones múltiples resultan difíciles de interpretar y producen imágenes ruidosas. Por otro lado, a pesar de que las proyecciones de grafos propuestas pueden mostrar información sobre la estructura de *clusters* de los datos, otros métodos de visualización de *clusters* generalmente producen imágenes más claras cuando el MAO no es muy pequeño. No obstante, los grafos pueden ser utilizados como alternativa cuando el MAO es pequeño (contiene menos de 100 neuronas aproximadamente), ya que en éstos casos otros métodos de análisis visual suelen presentar problemas. Por otro lado, las proyecciones de los grafos no están limitadas a MAO bidimensionales (nótese que un MAO unidimensional o bidimensional puede ser curvado *a posteriori* para poder apreciar todas las aristas del grafo, mientras que uno tridimensional requeriría una técnica de visualización mucho más sofisticada).

Finalmente, el grafo Hebbiano aporta imágenes más precisas en cuanto a visualizar el error topográfico (y por tanto de la preservación de la topología) y en cuanto al análisis de *clusters*. Sin embargo, el AGM y el grafo que conecta los  $K$  primeros vecinos de cada vector de referencia no dependen del número de muestras de la distribución. Su complejidad computacional es  $\mathcal{O}(M^2)$  mientras la del grafo Hebbiano es  $\mathcal{O}(MN)$ , donde  $M$  es el número de neuronas del MAO y

$N$  es el número de muestras de la distribución. De esta manera, pueden ser utilizados si la distribución de datos es muy grande y el MAO es de tamaño medio o pequeño, por razones de velocidad y para simplificar las imágenes reduciendo el número de aristas. Por otro lado, la determinación del grafo más adecuado para evaluar si un MAO se ha adaptado a los datos “lo mejor posible” depende de las características de la distribución, que son desconocidas *a priori*.

## Capítulo 7

# CONCLUSIONES, APORTACIONES Y LÍNEAS FUTURAS

### 7.1. Conclusiones y Aportaciones

Siguiendo la filosofía del AED, donde un conjunto de datos debe ser explorado de tantas maneras como sea posible, esta tesis propone cuatro nuevos métodos de análisis visual de MAO como principal aportación, además de llevar a cabo una extensa recopilación y análisis de los métodos de visualización más relevantes descritos en la literatura. Los nuevos métodos están basados en estrategias diferentes para poder tratar varios tipos de distribuciones de datos y MAO de diferentes tamaños, para los cuales un determinado método de visualización pueda resultar más idóneo. Los nuevos métodos revelan diversas características de los conjuntos de datos y presentan mejores prestaciones en comparación con otros métodos de análisis visual descritos en la literatura. En definitiva, los nuevos métodos complementan, mejoran y facilitan el análisis visual de MAO, tanto desde un punto de vista exploratorio, como desde el punto de vista de analizar el proceso de adaptación del MAO a una distribución de datos.

#### 7.1.1. Conclusiones

Del análisis de los diferentes experimentos realizados y de los resultados obtenidos con cada método es posible derivar las siguientes conclusiones:

- Método de semejanza de triángulos:
  - Su principal ventaja es la posibilidad de emplear más de tres neuronas para calcular la proyección que genera. Esto es fundamental para conseguir proyecciones continuas y robustas, y aportar descripciones adecuadas de conjuntos de datos. En este sentido, es posible apreciar un aumento considerable en la calidad de las imágenes con respecto a otros métodos de interpolación descritos en la literatura.
  - Al ser capaz de proyectar cada dato de la distribución a un espacio observable, la resolución y detalle de las imágenes es superior a la de otros métodos basados en conjuntos de vectores de referencia.
  - El método también puede desvincularse de la estructura regular de las neuronas en los mapas, lo cual permite ilustrar estructuras difíciles de observar con la mayoría de métodos de visualización, y es robusto a la adaptación del MAO en cuanto al error de cuantización.
  - Las proyecciones que mejor describen los datos son generadas empleando todos los pares de neuronas del MAO, para que resulten continuas. A pesar de que el MST puede generar descripciones adecuadas de los datos con MAO grandes, la velocidad del método disminuye al emplear un número elevado de pares de neuronas, por lo que el tamaño del MAO debe ser moderado. Por otro lado, el método resulta adecuado cuando el MAO es pequeño ya que aporta la resolución necesaria que no puede proporcionar un conjunto reducido de vectores de referencia.
- Superposición de imágenes:
  - Genera imágenes más completas y robustas que las dos originales cuando ambas representan el mismo tipo de información. Su principal ventaja reside en que resulta mucho más fácil comparar imágenes similares cuando éstas son superpuestas una encima de otra, que compararlas ubicándolas una al lado de la otra.
  - La interpretación de dos colores no es complicada, aunque un tercer color puede incorporar demasiada complejidad a la imagen.
  - Al combinar matrices de distancias con histogramas de datos el método produce imágenes mejoradas y más robustas con las cuales se facilita la

comprensión y el análisis de la distribución con la que se ha entrenado el MAO.

- La superposición de imágenes es más idónea cuando el tamaño del MAO es elevado.
- Algoritmo de agrupación de neuronas:
  - Los experimentos llevados a cabo muestran como los métodos de contracción de neuronas mejoran notablemente el proceso de análisis de un MAO con respecto a los métodos tradicionales. Aportan una proyección no-lineal de los vectores de referencia y un mapa coloreado, los cuales proporcionan diversa información útil sobre el MAO y sobre la distribución con la que ha sido entrenado.
  - Con respecto al método de Himberg, el AAN produce descripciones menos precisas dada su naturaleza aleatoria y los conjuntos reducidos de neuronas con los que trabaja. La ventaja sobre el método de Himberg reside en el aumento de velocidad cuando el tamaño del MAO es elevado, mientras las imágenes generadas siguen siendo similares a las obtenidas con el método de Himberg. La mayor velocidad del AAN también es crucial de cara a analizar el proceso dinámico de contracción de neuronas cuando el tamaño del MAO es elevado.
  - En la práctica, los métodos de contracción no producen resultados interesantes con MAO pequeños. Para los de tamaño medio, el método de Himberg produce los resultados más precisos prácticamente en tiempo real. Por tanto, el AAN debe ser aplicado a MAO de tamaño relativamente elevado (más de mil neuronas), sobre todo si el objetivo es analizar el proceso dinámico de contracción, dada su complejidad computacional más ligera.
- Proyección de grafos:
  - La proyección del grafo Hebbiano representa una versión gráfica de varias medidas de bondad o preservación de la topología al estar íntimamente relacionada con el error topográfico y la medida Hebbiana. Los experimentos muestran como esta proyección indica correctamente las discontinuidades presentes en el MAO y si la dimensión intrínseca

de los datos asociados a cada región del MAO es menor, igual o mayor que su propia dimensión.

- En la mayoría de experimentos llevados a cabo (incluido el basado en datos reales) el subgrafo Hebbiano que conecta neuronas vecinas, el AGM y el grafo de los  $K$ -vecinos más cercanos para valores bajos de  $K$  permiten analizar la calidad de la adaptación del MAO a una distribución.
  - Los grafos basados en el principio de aprendizaje Hebbiano competitivo suelen generar imágenes relacionadas con otros métodos de visualización de MAO al revelar las neuronas interpolantes. De esta manera, también pueden ser utilizados para observar estructuras como *clusters*, especialmente si el MAO es pequeño, donde tienen problemas la mayoría de métodos de visualización.
  - La proyección de grafos resulta más idónea cuando el tamaño del MAO no es elevado, ya que un gran número de aristas puede dificultar la interpretación de las imágenes.
- Relaciones entre los nuevos métodos:
    - Las matrices de distancias y los histogramas de datos proporcionan información exacta y precisa sobre distancias y proporciones de datos en el MAO, por lo que su superposición no está basada en una estrategia heurística, a diferencia del AAN. Sin embargo, en la práctica los métodos de contracción proporcionan más información útil por lo que sus prestaciones son notablemente mayores.
    - La diferencia principal entre el MST y el AAN reside en que el primero proyecta patrones de datos, mientras el segundo proyecta un conjunto de vectores de referencia. De esta manera, el MST generalmente produce imágenes de mayor resolución. Ambos métodos pueden generar imágenes similares cuando el MAO contiene un número elevado de neuronas. Sin embargo, en esos casos el MST necesita utilizar un número elevado de pares de neuronas para generar proyecciones precisas, por lo que resulta mucho más lento que el AAN.
    - El encadenado basado en un código de colores de la proyección generada con el AAN con la original del MAO asigna una serie de colores a los

vectores de referencia que pueden ser utilizados en otras proyecciones de los datos o de los propios vectores de referencia. De esta manera, es posible mejorar el MST combinándolo con el AAN para producir proyecciones coloreadas de los datos de entrada.

- Aunque en algunos casos es posible detectar discontinuidades en los MAO mediante los mapas coloreados generados con el AAN, resulta mucho más adecuado utilizar las proyecciones de grafos para analizar la preservación de la topología del MAO. Por otro lado, ambos métodos pueden ser combinados mediante una superposición para corroborar sus resultados.
- El propósito de la estrategia de proyección de grafos es aportar una medida visual e intuitiva de la preservación de la topología en varias regiones del MAO. De esta manera, las visualizaciones no pueden ser comparadas con las generadas por el resto de métodos descritos en esta tesis, aunque en algunos casos existe una relación entre las imágenes que generan.
- A pesar de que las proyecciones de grafos propuestas pueden mostrar información sobre la estructura de *clusters*, la mayoría de métodos de visualización de *clusters* producen imágenes más claras, especialmente cuando el MAO no es muy pequeño.

### 7.1.2. Aportaciones

Además de la extensa recopilación y descripción de los métodos de visualización de MAO más relevantes de la literatura, la principal aportación de la tesis consiste en la descripción y análisis de cuatro nuevos métodos de visualización de MAO:

- Método de semejanza de triángulos:
  - Se trata de un método mejorado, sofisticado y robusto de interpolación que proyecta cada dato de la distribución a un espacio observable en función de unos puntos de referencia y un criterio geométrico.
  - A diferencia de la mayoría de métodos de visualización de MAO, aporta información sobre cada dato individual de la distribución mediante una visualización explícita de cada patrón. De esta manera, resulta



atractivo desde el punto de vista del AED, ya que puede proyectar todos los datos de la distribución de manera no-lineal.

- Es un método general de proyección no-lineal. No está limitado a la correspondencia discreta que genera un MAO.
- Superposición de imágenes:
  - Es una solución sencilla que ayuda a la interpretación de la información contenida en los MAO, a pesar de que el método no añade información nueva, ya que las imágenes a combinar pueden ser examinadas individualmente.
  - Aplicado a matrices de distancias y a histogramas de datos, las imágenes resultantes facilitan un análisis más completo y robusto de la estructura de *clusters* de la distribución con la que se ha entrenado un MAO.
- Algoritmo de agrupación de neuronas:
  - Es un algoritmo eficiente de contracción de neuronas, el cual está basado en una heurística distinta a la empleada por métodos gravitacionales, que considera subconjuntos reducidos de neuronas para actualizar sus posiciones por medio de un algoritmo auto-organizativo similar al del entrenamiento secuencial del MAO.
  - El método muestra como el algoritmo del MAO puede ser visto desde un punto de vista más general para poder ser aplicado a otros problemas diferentes del entrenamiento de MAO. En el AAN los conceptos son invertidos para conseguir organizar neuronas en el mapa en función de la organización de sus vectores de referencia en el espacio de los datos.
  - El AAN representa una seria alternativa a los métodos tradicionales de análisis de *clusters* ya que aporta más información útil. Genera una proyección no-lineal de los vectores de referencia y un mapa coloreado que permite detectar los diferentes *clusters* y los bordes entre ellos, facilita el análisis de trayectorias y el etiquetado de las neuronas, y puede aportar una medida visual de bondad del MAO.

- Proyección de grafos:
  - Representa la única estrategia visual hasta la fecha para analizar la preservación de la topología y la calidad de la adaptación del MAO a la distribución de datos.
  - Los grafos propuestos también pueden revelar información sobre la estructura de *clusters* de la distribución.

## 7.2. Líneas Futuras

Considerando las conclusiones y aportaciones comentadas, las líneas futuras del presente trabajo pueden estar dirigidas en las siguientes direcciones:

- Método de semejanza de triángulos:
  - Estudiar otros tipos de errores entre triángulos, por ejemplo, basados en los lugares geométricos descritos en las Figuras 6.3 y 6.4.
  - Estudiar la velocidad del algoritmo empleando un método de optimización más eficiente.
  - Compararlo con otros métodos de proyección no-lineales.
  - Estudiar su aplicación con otras proyecciones de vectores de referencia, distintas a la regular del MAO.
  - Estudiar aspectos teóricos del algoritmo.
- Superposición de imágenes:
  - Estudiar la idoneidad del uso de colores en la superposición, del modelo RGB y de otros modelos de color uniformes.
  - Estudiar la idoneidad de utilizar un par de colores determinado.
  - Estudiar qué tipo de información puede ser combinada, mediante un tercer color, con una matriz de distancias y un histograma de datos para mejorar el análisis de la estructura de *clusters* de las distribuciones.
- Algoritmo de agrupación de neuronas:

- Estudiar nuevas estrategias (por ejemplo, ciclos aleatorios o enfoques no-aleatorios) para seleccionar las neuronas elegidas, de cara a obtener un mayor grado de precisión en las proyecciones.
- Estudiar variantes del método de contracción de Himberg más eficientes que actualicen un número reducido de neuronas en cada iteración. Comparar los resultados de estas variantes con el AAN.
- Proyección de grafos:
  - Estudiar estrategias para generar grafos coloreados.
  - Estudiar proyecciones de otros tipos de grafos, por ejemplo, basados en nuevos índices que indiquen la “gravedad” de las discontinuidades en el MAO.
- Otras líneas futuras:
  - Estudiar la combinación de los métodos propuestos.
  - Estudiar la aplicación de los métodos de visualización a variantes de los MAO.
  - Diseñar variantes de los MAO con el objetivo de mejorar el proceso de análisis exploratorio.
  - Adaptar los métodos de visualización para poder analizar MAO tridimensionales.
  - Diseñar nuevos métodos de visualización para tratar MAO tridimensionales o incluso de mayor dimensión.

# Apéndice A

## SÍMBOLOS, ACRÓNIMOS Y TRADUCCIÓN DE TÉRMINOS

### A.1. Símbolos

La siguiente lista ofrece una breve descripción de los símbolos más relevantes utilizados en la tesis<sup>1</sup>:

$C_k$	clase $k$ asociada a una distribución de datos
$I(x)$	intensidad luminosa asociada a la variable $x$
$\mathcal{I}_r$	imagen monocromática expresada en tonos de rojo
$\mathcal{I}_g$	imagen monocromática expresada en tonos de verde
$\mathcal{I}_b$	imagen monocromática expresada en tonos de azul
$\mathcal{I}_C(x)$	color del pixel $x$ de una imagen combinada
$\mathcal{I}_U$	imagen de una U-matrix
$\mathcal{I}_H$	imagen de un histograma de datos
$K$	número de clases de una distribución de datos
$M$	número de neuronas de un MAO
$N$	número de vectores de entrada de una distribución de datos
$N_k$	número de muestras pertenecientes a la clase $C_k$
$S$	matriz de similitudes normalizada del método de Himberg
$\tilde{S}$	matriz de similitudes del método de Himberg
$T$	número total de iteraciones de un algoritmo
$V_k$	región de Voronoi asociada a la clase $C_k$
$X$	número de filas de un MAO bidimensional
$X_i$	posiciones de las neuronas en la iteración $i$ del método de Himberg
$Y$	número de columnas de un MAO bidimensional

---

<sup>1</sup>Algunas definiciones de los símbolos pueden variar dependiendo del contexto en el que sean utilizados.

## A.2. ACRÓNIMOS

---

$c$	índice que representa a la neurona ganadora
$d$	distancia
$f$	proyección del espacio de los datos al espacio de observación
$f_d$	correspondencia discreta entre $\Omega$ y $\Phi$
$h_{c,i}(t)$	núcleo de vecindad del algoritmo secuencial del MAO
$n$	dimensión del espacio de los datos
$m$	dimensión del espacio de observación
$m_k$	vector de referencia asociado a la clase $C_k$
$p$	proyección del vector de entrada $x$ sobre el espacio de observación
$p(x)$	densidad de probabilidad de una distribución de datos
$r$	posición de una neurona o iteraciones del método de Himberg
$t$	instante de tiempo correspondiente a una iteración de un algoritmo
$x$	vector de entrada de una distribución de datos
$w$	vector de referencia
$\Delta$	distribución de datos de entrenamiento
$\Xi$	conjunto de neuronas de un MAO
$\Xi_c$	conjunto de neuronas alrededor de la ganadora
$\Upsilon(x)$	conjunto de neuronas utilizado para formar $\Psi(x)$
$\Phi$	conjunto de posiciones de las neuronas de un MAO
$\Psi(x)$	conjunto de pares de neuronas asociados a $x$
$\Omega$	conjunto de vectores de referencia de un MAO
$\alpha(t)$	tasa de aprendizaje del algoritmo secuencial del MAO
$\beta(t)$	función lineal decreciente utilizada en $\psi_{c,i}(t)$
$\varepsilon_x(p)$	coste ponderado al proyectar $x$ sobre $p$
$\eta$	neurona de un MAO
$\theta$	parámetro que controla la fuerza de atracción en el método de Himberg
$\lambda(t)$	tasa de aproximación del AAN
$\xi$	distancia o error entre triángulos
$\sigma(t)$	radio de vecindad del núcleo de vecindad de un algoritmo
$\tau$	porcentaje del diámetro de $\Omega$
$\varphi$	función de ponderación utilizada en $\varepsilon_x(p)$
$\chi_{c,i}(t)$	núcleo de vecindad binario centrado en la neurona $\eta_c$
$\psi_{c,i}(t)$	núcleo de vecindad del AAN

## A.2. Acrónimos

### A.2.1. En español

AAN	Algoritmo de Agrupación de Neuronas
ACP	Análisis de Componentes Principales
AED	Análisis Exploratorio de Datos
AGM	Árbol Generador Mínimo

CV	Cuantización Vectorial
EMD	Escalado Multidimensional
IA	Inteligencia Artificial
MAO	Mapa Auto-Organizativo
MST	Método de Semejanza de Triángulos
RNA	Red Neuronal Artificial
UMS	Unidad de Máxima Similitud

### A.2.2. En inglés

<i>CRISP-DM</i>	<i>CRoss Industry Standard Process model for Data Mining</i>
<i>LMS</i>	<i>Least Mean Square</i>
<i>LPC</i>	<i>Linear Predictive Coding</i>
<i>PLP</i>	<i>Perceptual Linear Prediction</i>
<i>RASTA-PLP</i>	<i>RelAtive SpecTrAl-Perceptual Linear Prediction</i>

### A.3. Traducción de términos

Algoritmo de agrupación de neuronas (AAN)	<i>Grouping neuron algorithm (GNA)</i>
Análisis confirmatorio de datos	<i>Confirmatory data analysis (CDA)</i>
Análisis de componentes curvilíneos	<i>Curvilinear component analysis (CCA)</i>
Análisis de componentes independientes	<i>Independent component analysis (ICA)</i>
Análisis de componentes principales (ACP)	<i>Principal component analysis (PCA)</i>
Análisis de discriminación lineal	<i>Linear discriminant analysis (LDA)</i>
Análisis exploratorio de datos (AED)	<i>Exploratory data analysis (EDA)</i>
Aprendizaje automático	<i>Machine learning (ML)</i>
Árbol generador mínimo (AGM)	<i>Minimal spanning tree (MST)</i>
Característica invariante	<i>Invariant feature</i>
Centroide	<i>Centroid</i>
Clasificación automática	<i>Clustering<sup>2</sup></i>
Código de colores	<i>Color coding</i>
Coefficiente de correlación de clasificación	<i>Rank correlation coefficient</i>
Coefficiente de Spearman	<i>Spearman's coefficient</i>
Conexiones entre <i>clusters</i>	<i>Cluster connections</i>
Conglomerado	<i>Cluster<sup>2</sup></i>
Coordenadas adaptativas	<i>Adaptive coordinates</i>
Cuantización vectorial (CV)	<i>Vector quantization (VQ)</i>
Cuantización vectorial de aprendizaje	<i>Learning vector quantization (LVQ)</i>
Curvas principales	<i>Principal curves</i>

---

<sup>2</sup>Los términos ingleses *cluster* y *clustering* han sido incluidos en el texto ya que su uso está muy extendido.

### A.3. TRADUCCIÓN DE TÉRMINOS

---

Descubrimiento de conocimiento en bases de datos	– <i>Knowledge discovery in databases</i>
Diagrama de dispersión	– <i>Scatter plot</i>
Elemento de proceso	– <i>Processing element</i>
Empotrados locales	– <i>Local embeddings</i>
Encadenado	– <i>Link</i>
Error de cuantización medio	– <i>Average quantization error</i>
Error topográfico	– <i>Topographic error</i>
Escalado multidimensional (EMD)	– <i>Multidimensional scaling (MDS)</i>
Espacio de características	– <i>Feature space</i>
Espacio de entrada	– <i>Input space</i>
Espacio de los datos	– <i>Data space</i>
Espacio de observación	– <i>Observation space</i>
Espacio de salida	– <i>Output space</i>
Función topográfica	– <i>Topographic function</i>
Funciones de base radial	– <i>Radial basis functions (RBF)</i>
Grafo Hebbiano	– <i>Hebb graph</i>
Hipersuperficie	– <i>Manifold</i>
Histograma de datos	– <i>Data histogram</i>
Histogramas de datos suaves	– <i>Smooth data histograms (SDH)</i>
Integrador con pérdidas	– <i>Leaky integrator</i>
Inteligencia artificial (IA)	– <i>Artificial intelligence (AI)</i>
Maldición de la dimensionalidad	– <i>Curse of dimensionality</i>
Mapa auto-organizativo (MAO)	– <i>Self-organizing map (SOM)</i>
Mapa auto-organizativo de subespacio adaptativo	– <i>Adaptive subspace self-organizing map (ASSOM)</i>
Mapa auto-organizativo recurrente	– <i>Recurrent self-organizing map (RSOM)</i>
Mapa de Kohonen temporal	– <i>Temporal Kohonen map (TKM)</i>
Mapa por lotes	– <i>Batch map</i>
Marco de alineación	– <i>Alignment framework</i>
Matriz de dispersión	– <i>Scatter plot matrix</i>
Matriz de distancia unificada	– <i>Unified distance matrix</i>
Matriz de distancias	– <i>Distance matrix</i>
Máxima verosimilitud	– <i>Maximum likelihood</i>
Medida de bondad	– <i>Goodness measure</i>
Medida de Zrehen	– <i>Zrehen measure</i>
Medida Hebbiana	– <i>Hebb measure</i>
Medida-C	– <i>C-measure</i>
Método de semejanza de triángulos (MST)	– <i>Triangle similarity method (TSM)</i>
Minería de datos	– <i>Data mining</i>
Neurona	– <i>Neuron</i>
Neurona ganadora	– <i>Winner neuron</i>
Neurona interpolante	– <i>Interpolating neuron</i>

Nodo – *Node*  
Núcleo de vecindad – *Neighborhood kernel*  
Orden de clasificación – *Rank order*  
Perceptrón multicapa – *Multi-layer perceptron (MLP)*  
Plano de componente – *Component plane*  
Predicción lineal perceptiva – *Perceptual linear prediction (PLP)*  
Preservación de la topología – *Topology preservation*  
Principio de aprendizaje Hebbiano competitivo – *Competitive Hebbian learning principle*  
Producto topográfico – *Topographic product*  
Proyección de Sammon – *Sammon's mapping*  
Proyección topográfica generativa – *Generative topographic mapping (GTM)*  
Red neuronal artificial (RNA) – *Artificial neural network (ANN)*  
Red neuronal de pulsos – *Pulsed neural network*  
Red neuronal de tiempo-retrasado – *Time-delay neural network (TDNN)*  
Redes de representación topológica – *Topology representing networks*  
Rejilla de distancias – *Distance grid*  
Retropropagación del error – *Error backpropagation*  
Sobreajuste – *Overfitting*  
Teoría de resonancia adaptativa – *Adaptive resonance theory (ART)*  
Triangulación inducida de Delaunay – *Induced Delaunay triangulation*  
Tubo de rayos catódicos – *Cathode ray tube (CRT)*  
Unidad – *Unit*  
Unidad de máxima similitud (UMS) – *Best matching unit (BMU)*  
Vector de pesos – *Weight vector*  
Vector de referencia – *Reference vector*  
Vector del diccionario de bloques – *Codebook vector*  
Unidad binaria de umbral – *Binary threshold unit*



# Bibliografía

- [AG00] C. Ambroise y G. Govaert. Clustering by maximizing a fuzzy classification maximum likelihood criterion. En J. Bethlehem y P van der Heijden, eds., *Compstat 2000, Proceedings in Computational Statistics, 14th Symposium*, págs. 186–192. Utrecht, The Netherlands, agosto 2000.
- [AHS85] D. H. Ackley, G. Hinton y T. Sejnowski. A learning algorithm for Boltzmann machines. *Cognitive Science*, tomo 9: págs. 147–169, 1985.
- [AHV99] Esa Alhoniemi, Johan Himberg y Juha Vesanto. Probabilistic measures for responses of self-organizing map units. En *Proceedings of the International ICSC Congress on Computational Intelligence Methods and Applications (CIMA'99)*, págs. 286–290. ICSC Academic Press, 1999. URL [lib.hut.fi/Diss/2002/isbn9512258978/](http://lib.hut.fi/Diss/2002/isbn9512258978/).
- [AKCM90] Stanley C. Ahalt, Ashok K. Krishnamurthy, Prakoon Chen y Douglas E. Melton. Competitive learning algorithms for vector quantization. *Neural Networks*, tomo 3: págs. 277–290, 1990.
- [And72] D. F. Andrews. Plots of high-dimensional data. *Biometrics*, tomo 28: págs. 125–136, 1972.
- [AR88] J. A. Anderson y E. Rosenfeld, eds. *Neurocomputing: Foundations of Research*. MIT Press, Cambridge Mass., 1988.
- [ASRJ77] J. A. Anderson, J. W. Silverstein, S. A. Ritz y R. S. Jones. Distinctive features, categorical perception, and probability learning: some applications of a neural model. *Psychological Review*, tomo 84: págs. 413–451, 1977.
- [Beh97] John T. Behrens. Principles and procedures of exploratory data analysis. *Psychological Methods*, tomo 2(2): págs. 131–160, 1997.
- [Ber83] Jacques Bertin. *The Semiology of Graphics*. University of Wisconsin Press, 1983.
- [Bez74] James C. Bezdek. Numerical taxonomy with fuzzy sets. *Journal of Mathematical Biology*, tomo 1(1): págs. 57–71, 1974.
- [BGO97] Matthias Burger, Thore Graepel y Klaus Obermayer. Phase transitions in stochastic self-organizing maps. *Physical Review E*, tomo 56(4): págs. 3876–3890, 1997. URL [stat.cs.tu-berlin.de/~burg/publications.html](http://stat.cs.tu-berlin.de/~burg/publications.html).

- [BH65] G. H. Ball y D. J. Hall. Isodata: a novel method of data analysis and pattern classification. Informe Técnico AD 699616, Stanford Research Institute, Menlo Park, CA, 1965.
- [BHV99] Hans-Ulrich Bauer, Michael Herrmann y Thomas Villmann. Neural maps and topographic vector quantization. *Neural Networks*, tomo 12(4-5): págs. 659–676, 1999. URL [www.gwdg.de/~mherrma/paper.html](http://www.gwdg.de/~mherrma/paper.html).
- [Bis95] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, 1995.
- [Bis98] Christopher M. Bishop, ed. *Neural Networks and Machine Learning*. Springer, 1998.
- [BJD81] G. Biswas, A. K. Jain y R. C. Dubes. Evaluation of projection algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tomo PAMI-3(6): págs. 701–708, noviembre 1981.
- [BKOS97] M. de Berg, M. van Kreveld, M. Overmars y O. Schwarzkopf. *Computational Geometry, Algorithms and Applications*. Springer, 1997.
- [BL88] D. S. Broomhead y D. Lowe. Multivariable functional interpolation and adaptive networks. *Complex Systems*, tomo 2: págs. 321–355, 1988.
- [BP92] Hans-Ulrich Bauer y Klaus R. Pawelzik. Quantifying the neighborhood preservation of self-organizing feature maps. *IEEE Transactions on Neural Networks*, tomo 3(4): págs. 570–579, 1992. URL [www.cnl.salk.edu/~wiskott/Bibliographies/NeuralMaps.html](http://www.cnl.salk.edu/~wiskott/Bibliographies/NeuralMaps.html).
- [BP95] James C. Bezdek y Nikhil R. Pal. An index of topological preservation for feature extraction. *Pattern Recognition*, tomo 28(3): págs. 381–391, 1995.
- [Bre03] Isaac Brewer. *Overview of Pre-computer Cartographic Visualization*, 2003. URL [www.cartographica.com:10000/Visualization.htm](http://www.cartographica.com:10000/Visualization.htm). Documento Electrónico.
- [BS95] A. J. Bell y T. J. Sejnowski. An information maximisation approach to blind separation and blind deconvolution. *Neural Computation*, tomo 7(6): págs. 1129–1159, 1995. URL [www.cnl.salk.edu/~tony/ica.html](http://www.cnl.salk.edu/~tony/ica.html).
- [BSW98] Christopher M. Bishop, Markus Svensen y Christopher K. I. Williams. GTM: The generative topographic mapping. *Neural Computation*, tomo 10(1): págs. 215–234, 1998. URL [www.ncrg.aston.ac.uk/GTM/](http://www.ncrg.aston.ac.uk/GTM/).
- [Cai61] E. R. Caianiello. Outline of a theory of thought processes and thinking machines. *Theoretical Biology*, tomo 2: págs. 204–235, 1961.
- [Car99] Otávio Augusto S. Carpinteiro. A hierarchical self-organizing map model for sequence recognition. *Neural Processing Letters*, tomo 9(3): págs. 209–220, 1999. URL [www.iee.efei.br/~otavio/tmp/npl.pdf](http://www.iee.efei.br/~otavio/tmp/npl.pdf).
- [Cas96] K. R. Castleman. *Digital Image Processing*. Prentice Hall, Englewood Cliffs, NJ, 1996.

- [CCK<sup>+</sup>99] Pete Chapman, Julian Clinton, Thomas Khabaza, Thomas Reinartz y Rüdiger Wirth. The CRISP-DM process model. Informe técnico, CRISP-DM consortium, marzo 1999. URL [www.crisp-dm.org](http://www.crisp-dm.org).
- [CFP94] M. Cottrell, J. C. Fort y G. Pagès. Two or three things that we know about the Kohonen algorithm. En M. Verleysen, ed., *Proc. ESANN'94, European Symp. on Artificial Neural Networks*, págs. 235–244. D facto conference services, Brussels, Belgium, 1994. URL [www.dice.ucl.ac.be/esann/proceedings/esann1994/content.htm](http://www.dice.ucl.ac.be/esann/proceedings/esann1994/content.htm).
- [CFP98] M. Cottrell, J.C. Fort y G. Pagès. Theoretical aspects of the SOM algorithm. *Neurocomputing*, tomo 21: págs. 119–138, 1998.
- [CG91] G. A. Carpenter y S. Grossberg. *Pattern recognition by self-organizing neural networks*. MIT Press, Cambridge Mass., 1991.
- [Che73] H. Chernoff. The use of faces to represent points in k-dimensional space graphically. *Journal of the American Statistical Association*, tomo 68: págs. 361–368, 1973.
- [Com94] Pierre Comon. Independent component analysis, a new concept? *Signal Processing*, tomo 36(3): págs. 287–314, abril 1994. URL [www.i3s.unice.fr/~comon/SP94.html](http://www.i3s.unice.fr/~comon/SP94.html).
- [CT93] Geoffrey J. Chappell y John G. Taylor. The temporal Kohonen map. *Neural Networks*, tomo 6: págs. 441–445, 1993.
- [CU93] A. Cichocki y R. Unbehauen. *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons, 1993.
- [Día99a] Francisco Díaz. *Utilización de métodos discriminantes en el modelado acústico de la voz*. Tesis Doctoral, Universidad Politécnica de Madrid, 1999.
- [Día99b] V. Díaz. *Técnicas de análisis de datos para investigadores sociales. Aplicaciones prácticas con SPSS para Windows*. Rama, MADrid, 1999.
- [Dav02] Ian Davidson. Visualizing clustering results. En *Proceedings of the Second SIAM International Conference on Data Mining*. 2002. URL [www.siam.org/meetings/sdm02/proceedings/sdm02-01.pdf](http://www.siam.org/meetings/sdm02/proceedings/sdm02-01.pdf).
- [DF84] P. Diaconis y D. Freedman. Asymptotics of graphical projection pursuit. *Annals of Statistics*, tomo 12: págs. 793–815, 1984.
- [DH97] P. Demartines y J. Hérault. Curvilinear component analysis: a self-organising neural network for non-linear mapping of data sets. *IEEE Trans. on Neural Networks*, tomo 8(1): págs. 148–154, 1997.
- [DM90] R. Durbin y G. Mitchison. A dimension reduction framework for understanding cortical maps. *Nature*, tomo 343: págs. 644–647, 1990.
- [EOS92] E. Erwin, K. Obermayer y K. J. Schulten. Self-organizing maps: Ordering, convergence properties and energy functions. *Biological Cybernetics*, tomo 67(1): págs. 47–55, 1992. URL [kon.cs.tu-berlin.de/publications/](http://kon.cs.tu-berlin.de/publications/).

- [EP96] N. R. Euliano y J. C. Principe. Spatio-temporal self-organizing feature maps. En *ICNN 96. The 1996 IEEE International Conference on Neural Networks (Cat. No. 96CH35907)*, tomo 4, págs. 1900–5. IEEE, New York, NY, USA, 1996. URL [www.cnel.ufl.edu/bib/publicationslist.html](http://www.cnel.ufl.edu/bib/publicationslist.html).
- [Fai98] Mark D. Fairchild. *Color Appearance Models*. Addison Wesley, Reading, Massachusetts., 1998.
- [Fau94] Laurene V. Fausett, ed. *Fundamentals of Neural Networks*. Prentice Hall, New York, 1994.
- [FH02] J. J. Filliben y A. Heckert. *Engineering Statistics Handbook: Exploratory Data Analysis*. NIST/SEMATECH, 2002. URL [www.itl.nist.gov/div898/handbook/eda/eda.htm](http://www.itl.nist.gov/div898/handbook/eda/eda.htm). Documento Electrónico.
- [Fis36] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, tomo 7(2): págs. 179–188, 1936.
- [Fle97] Arthur Flexer. Limitations of self-organizing maps for vector quantization and multidimensional scaling. En M. C. Mozer, M. I. Jordan y T. Petsche, eds., *Advances in Neural Information Processing Systems 9. Proceedings of the 1996 Conference*, págs. 445–451. MIT Press, London, UK, 1997. URL [www.ai.univie.ac.at/~arthur/](http://www.ai.univie.ac.at/~arthur/).
- [Fle01] Arthur Flexer. On the use of self-organizing maps for clustering and visualization. *Intelligent Data Analysis*, tomo 5(5): págs. 373–384, 2001. URL [www.ai.univie.ac.at/~arthur/](http://www.ai.univie.ac.at/~arthur/).
- [FMF<sup>+</sup>89] J. K. Flanagan, D. R. Morrell, R. L. Frost, C. J. Read y B. E. Nelson. Vector quantization codebook generation using simulated annealing. En *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, tomo 3, págs. 1759–1762. mayo 1989.
- [FO71] K. Fukunaga y D.R. Olsen. An algorithm for finding intrinsic dimensionality of data. *IEEE Transactions on Computers*, tomo c-20(2): págs. 176–183, 1971.
- [FP97] Craig L. Fancourt y Jose C. Principe. Temporal self-organization through competitive prediction. En *Proc. ICASSP'97*, tomo 4, págs. 3325–3329. 1997. URL [www.cnel.ufl.edu/bib/pdf\\_papers/fancourt97icassp.pdf](http://www.cnel.ufl.edu/bib/pdf_papers/fancourt97icassp.pdf).
- [Fri87] J. H. Friedman. Exploratory projection pursuit. *Journal of the American Statistical Association*, tomo 82(397): págs. 249–266, 1987.
- [Fri94] Bernd Fritzke. Growing cell structures – a self-organizing network for unsupervised and supervised learning. *Neural Networks*, tomo 7(9): págs. 1441–1460, 1994. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html).
- [Fri95a] Bernd Fritzke. Growing grid - a self-organizing network with constant neighborhood range and adaptation strength. *Neural Processing Letters*, tomo 2(5): págs. 9–13, 1995. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html).

- [Fri95b] Bernd Fritzke. A growing neural gas network learns topologies. En G. Tesauero, D. S. Touretzky y T. K. Leen, eds., *Advances in Neural Information Processing Systems 7*, págs. 625–632. MIT Press, Cambridge MA, 1995. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html).
- [Fri96] Bernd Fritzke. Unsupervised ontogenetic networks. En E. Fiesler y R. Beale, eds., *Handbook of Neural Computation*. Institute of Physics Publishing and Oxford University Press, 1996. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html).
- [Fri97] Bernd Fritzke. The LBG-U method for vector quantization - an improvement over LBG inspired from neural networks. *Neural Processing Letters*, tomo 5(1): págs. 35–45, 1997. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/PUB-LIST/newlist/newlist.html).
- [Fri03] Michael Friendly. *Gallery of Data Visualization, The Best and Worst of Statistical Graphics*, 2003. URL [www.math.yorku.ca/SCS/Gallery/intro.html](http://www.math.yorku.ca/SCS/Gallery/intro.html). Documento Electrónico.
- [FT74] J. H. Friedman y J. W. Tukey. A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computers*, tomo 23(9): págs. 881–889, 1974.
- [Gam01] Daniel Manrique Gamo. *Diseño de redes de neuronas y nuevas técnicas de optimización mediante algoritmos genéticos*. Tesis Doctoral, Universidad Politécnica de Madrid, marzo 2001.
- [GDN03] Jonatan Gomez, Dpankar Dasgupta y Olfa Nasraoui. A new gravitational clustering algorithm. En *Proc. 2003 SIAM Conference on Data Mining*. San Francisco, CA, abril 2003. URL [archer.ee.memphis.edu/www.ee.memphis.edu/people/faculty/nasraoui/publications.html](http://archer.ee.memphis.edu/www.ee.memphis.edu/people/faculty/nasraoui/publications.html).
- [Gol89] David E. Goldberg. *Genetic Algorithms in Search, Optimisation, and Machine Learning*. Addison-Wesley Publishing Company, Inc., 1989.
- [GR93] Josef Göppert y Wolfgang Rosenstiel. Topology preserving interpolation in self-organizing maps. En *Proc. NeuroNimes'93*, págs. 425–434. EC2, Nanterre, France, octubre 1993. URL [www-ti.informatik.uni-tuebingen.de/~goeppert/paper\\_abstract.htm](http://www-ti.informatik.uni-tuebingen.de/~goeppert/paper_abstract.htm).
- [GR95a] Josef Göppert y Wolfgang Rosenstiel. Interpolation in SOM: Improved generalization by iterative methods. En *Proc. of ICANN'95*, págs. 425–434. EC2 & Cie, Paris, Francia, octubre 1995. URL [www-ti.informatik.uni-tuebingen.de/~goeppert/paper\\_abstract.htm](http://www-ti.informatik.uni-tuebingen.de/~goeppert/paper_abstract.htm).
- [GR95b] Josef Göppert y Wolfgang Rosenstiel. Topological interpolation in SOM by affine transformations. En *Proc. of ESANN'95*. D Facto, Brussels, Belgium, abril 1995. URL [www-ti.informatik.uni-tuebingen.de/~goeppert/paper\\_abstract.htm](http://www-ti.informatik.uni-tuebingen.de/~goeppert/paper_abstract.htm).

- [Gre98] Marc Green. *Toward a Perceptual Science of Multidimensional Data Visualization: Bertin and Beyond*, 1998. URL [www.ergogero.com/dataviz/dviz0.html](http://www.ergogero.com/dataviz/dviz0.html). Documento Electrónico.
- [GS96] Geoffrey J. Goodhill y Terrence J. Sejnowski. Quantifying neighbourhood preservation in topographic mappings. En *Proceedings of the 3rd Joint Symposium on Neural Computation*, págs. 61–82. 1996. URL [cns.georgetown.edu/~geoff/pubs.html](http://cns.georgetown.edu/~geoff/pubs.html).
- [GS97a] Geoffrey J. Goodhill y Terrence J. Sejnowski. Objective functions for topography: a comparison of optimal maps. En John A. Bullinaria, David G. Glasspool y George Houghton, eds., *Proceedings of the Fourth Neural Computation and Psychology Workshop: Connectionist Representations*. Springer-Verlag, London, 1997. URL [cns.georgetown.edu/~geoff/pubs.html](http://cns.georgetown.edu/~geoff/pubs.html).
- [GS97b] Geoffrey J. Goodhill y Terrence J. Sejnowski. A unifying objective function for topographic mappings. *Neural Computation*, tomo 9: págs. 1291–1304, 1997. URL [cns.georgetown.edu/~geoff/pubs.html](http://cns.georgetown.edu/~geoff/pubs.html).
- [Hay99] Simon Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 1999.
- [HD95] Denis Hamad y Stephane Delsert. Nonlinear mapping procedures for unsupervised pattern classification. En *Proc. International Conference on Engineering Applications of Neural Networks (EANN '95)*, págs. 457–460. Otaniemi, Finland, agosto 1995.
- [HE96] Christopher G. Healey y James T. Enns. A perceptual colour segmentation algorithm. Informe Técnico TR-96-09, Department of Computer Science, University of British Columbia, Vancouver, BC, Canada, 1996. URL [www.csc.ncsu.edu/faculty/healey/abstract/pubs.html](http://www.csc.ncsu.edu/faculty/healey/abstract/pubs.html).
- [Hea96] Christopher G. Healey. *Effective Visualization of Large Multidimensional Datasets*. Tesis Doctoral, University of British Columbia, Vancouver, Canada, 1996. URL [www.csc.ncsu.edu/faculty/healey/abstract/pubs.html](http://www.csc.ncsu.edu/faculty/healey/abstract/pubs.html).
- [Heb49] Donald O. Hebb. *The Organization of Behaviour*. John Wiley & Sons, New York, 1949.
- [Her90] Hynek Hermansky. Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, tomo 87(4): págs. 1738–1752, abril 1990.
- [HH95] Shyh-Jier Huang y Chuan-Chang Hung. Genetic algorithms enhanced Kohonen's neural networks. En *IEEE international conference on neural networks (ICNN'95)*, tomo 2. 1995.
- [Him98] Johan Himberg. Enhancing the SOM-based data visualization by linking different data projections. En *Proceedings of 1st International Symposium IDEAL'98, Intelligent Data Engineering and Learning—Perspectives on Financial Engineering and*

- Data Mining*, págs. 427–434. Springer, Hong Kong, 1998. URL [www.cis.hut.fi/projects/ide/publications/fulldetails.html](http://www.cis.hut.fi/projects/ide/publications/fulldetails.html).
- [Him00] Johan Himberg. A SOM based cluster visualization and its application for false coloring. En *International Joint Conference in Neural Networks (IJCNN2000)*, tomo 3, págs. 587–592. 2000. URL [www.cis.hut.fi/projects/ide/publications/fulldetails.html](http://www.cis.hut.fi/projects/ide/publications/fulldetails.html).
- [HK97] Erkki Häkkinen y Pasi Koikkalainen. The neural data analysis environment. En *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, págs. 69–74. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, junio 1997. URL [www.cis.hut.fi/wsom97/progababstracts/38.html](http://www.cis.hut.fi/wsom97/progababstracts/38.html).
- [HKLK97] Timo Honkela, Samuel Kaski, Krista Lagus y Teuvo Kohonen. WEBSOM—self-organizing maps of document collections. En *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, págs. 310–315. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, junio 1997. URL [www.cis.hut.fi/wsom97/progababstracts/51.html](http://www.cis.hut.fi/wsom97/progababstracts/51.html).
- [HM94] Hynek Hermansky y Nelson Morgan. RASTA processing of speech. *IEEE Transactions on Speech and Audio Processing*, tomo 2(4): págs. 578–589, octubre 1994.
- [HMBK91] Hynek Hermansky, Nelson Morgan, Aruna Bayya y Phil Kohn. RASTA-PLP speech analysis. Informe Técnico TR-91-069, International Computer Science Institute, Berkeley, CA., diciembre 1991. URL [www.icsi.berkeley.edu/real/rasta.html](http://www.icsi.berkeley.edu/real/rasta.html).
- [HO00] Aapo Hyvärinen y Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, tomo 13(4-5): págs. 411–430, 2000. URL [www.cis.hut.fi/aapo/papers/IJCNN99\\_tutorialweb](http://www.cis.hut.fi/aapo/papers/IJCNN99_tutorialweb).
- [Hop82] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. En *Proc. Natl. Academy of Sciences USA*, tomo 79, págs. 2554–2558. abril 1982.
- [Hot33] H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, tomo 24: págs. 417–441 and 498–520, 1933.
- [HP99] Aapo Hyvärinen y Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural Networks*, tomo 12(3): págs. 429–439, 1999. URL [www.cis.hut.fi/projects/ica/nonlinearica/](http://www.cis.hut.fi/projects/ica/nonlinearica/).
- [HS89] Trevor Hastie y Werner Stuetzle. Principal curves. *Journal of the American Statistical Association*, tomo 84(406): págs. 502–516, 1989.
- [Hub85] P. J. Huber. Projection pursuit (with discussion). *Annals of Statistics*, tomo 13(2): págs. 435–475, 1985.
- [Hyv99] A. Hyvärinen. Survey on independent component analysis. *Neural Computing Surveys*, tomo 2: págs. 94–128, 1999. URL [www.cis.hut.fi/aapo/papers/NCS99web/](http://www.cis.hut.fi/aapo/papers/NCS99web/).

- [IKKK94] Jukka Iivarinen, Teuvo Kohonen, Jari Kangas y Samuel Kaski. Visualizing the clusters on the self-organizing map. En Christer Carlsson, Timo Järvi y Tapio Reponen, eds., *Proceedings of Conference on Artificial Intelligence Research in Finland*, 12, págs. 122–126. Helsinki University of Technology, Neural Networks Research Centre, Helsinki, Finland, 1994.
- [IVO] Sistema integrado de reconocimiento robusto de voz (IVORY, integrated voice recognition system). URL [tamarisco.datsi.fi.upm.es/PROJECTS/IVORY/ivory.html](http://tamarisco.datsi.fi.upm.es/PROJECTS/IVORY/ivory.html). Proyecto de Investigación financiado por el Programa Nacional para las Tecnologías de la Información y las Comunicaciones. Duración: 1/9/1996–31/8/1998.
- [JM95] Daniel L. James y Risto Miikkulainen. SARDNET: a self-organizing feature map for sequences. En G. Tesauro, D. Touretzky y T. Leen, eds., *Advances in Neural Information Processing Systems 7*, págs. 577–584. MIT Press, Cambridge, MA, USA, 1995. URL [nn.cs.utexas.edu/pages/publications/abstracts.html](http://nn.cs.utexas.edu/pages/publications/abstracts.html).
- [JMM96] Anil K. Jain, Jianchang Mao y K. M. Mohiuddin. Artificial neural networks: A tutorial. *IEEE Computer*, tomo 29(3): págs. 31–44, 1996. URL [www.computer.org/computer/co1996/r3031abs.htm](http://www.computer.org/computer/co1996/r3031abs.htm).
- [JS87] M.C. Jones y R. Sibson. What is projection pursuit? *Journal of the Royal Statistical Society*, tomo 150: págs. 1–36, 1987.
- [KA96] Nagendra Kumar y Andreas G. Andreou. On generalizations of linear discriminant analysis. Informe Técnico JHU/ECE-96-07, Johns Hopkins University, 1996. URL [www.clsp.jhu.edu/ws97/acoustic/reports/LDArep.pdf](http://www.clsp.jhu.edu/ws97/acoustic/reports/LDArep.pdf).
- [Kan94] Jari Kangas. *On the Analysis of Pattern Sequences by Self-Organizing Maps*. Tesis Doctoral, Helsinki University of Technology, Espoo, Finland, 1994. URL [www.cis.hut.fi/~jari/papers/papers.html](http://www.cis.hut.fi/~jari/papers/papers.html).
- [Kas97] Samuel Kaski. *Data Exploration Using Self-Organizing Maps*. Tesis Doctoral, Helsinki University of Technology, marzo 1997. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [KHKL96] Teuvo Kohonen, Jussi Hynninen, Jari Kangas y Jorma Laaksonen. SOM\_PAK: The Self-Organizing Map Program Package. Informe Técnico A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996. URL [www.cis.hut.fi/research/som-research/nnrc-programs.shtml](http://www.cis.hut.fi/research/som-research/nnrc-programs.shtml).
- [Kiv96] Kimmo Kiviluoto. Topology preservation in self-organizing maps. En *Proceedings of the International Conference on Neural Networks (ICNN'96)*, tomo 1, págs. 294–299. junio 1996. URL [www.cis.hut.fi/kkluoto/publications/icnn96.pdf](http://www.cis.hut.fi/kkluoto/publications/icnn96.pdf).
- [Kiv98] Kimmo Kiviluoto. Comparing 2d and 3d self-organizing maps in financial data visualization. En T. Yamakawa y G. Matsumoto, eds., *Proceedings of the 5th International Conference on Soft Computing and Information/Intelligent Systems*



- (IIZUKA '98), tomo 1 de *Methodologies for the Conception, Design and Application of Soft Computing*, págs. 68–71. World Scientific, Iizuka, Fukuoka, Japan, octubre 1998. URL [www.cis.hut.fi/kkluoto/publications/iizuka98.pdf](http://www.cis.hut.fi/kkluoto/publications/iizuka98.pdf).
- [KK96] Samuel Kaski y Teuvo Kohonen. Exploratory data analysis by the self-organizing map: Structures of welfare and poverty in the world. En Apostolos-Paul N. Refenes, Yaser Abu-Mostafa, John Moody y Andreas Weigend, eds., *Neural Networks in Financial Engineering*, págs. 498–507. World Scientific, Singapore, 1996. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [KK98] Samuel Kaski y Teuvo Kohonen. Tips for processing and color-coding of self-organizing maps. En Guido Deboeck y Teuvo Kohonen, eds., *Visual Explorations in Finance with Self-Organizing Maps*, págs. 195–202. Springer, London, 1998.
- [KKL90] Jari Kangas, Teuvo Kohonen y Jorma Laaksonen. Variants of self-organizing maps. *IEEE Transactions on Neural Networks*, tomo 1(1): págs. 93–99, marzo 1990.
- [KKLZ00] B. Kégl, A. Krzyzak, T. Linder y K. Zeger. Learning and design of principal curves. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, tomo 22(3): págs. 281–297, 2000. URL [www.code.ucsd.edu/~zeger/pubs.html](http://www.code.ucsd.edu/~zeger/pubs.html).
- [KKS92] Mika Kasslin, Jari Kangas y Olli Simula. Process state monitoring using self-organizing maps. En I. Aleksander y J. Taylor, eds., *Artificial Neural Networks, 2*, tomo II, págs. 1531–1534. North-Holland, Amsterdam, Netherlands, 1992.
- [KL96] Samuel Kaski y Krista Lagus. Comparing self-organizing maps. En C. von der Malsburg, W. von Seelen, J. C. Vorbruggen y B. Sendhoff, eds., *Proceedings of ICANN'96, International Conference on Artificial Neural Networks*, Lecture Notes in Computer Science, págs. 809–814. Springer, Berlin, 1996. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [KMJ95] Martin A. Kraaijveld, Jianchang Mao y Anil K. Jain. A non-linear projection method based on Kohonen's topology preserving maps. *IEEE Trans. Neural Networks*, tomo 6(3): págs. 548–559, mayo 1995.
- [KMSK91] Teuvo Kohonen, Kai Mäkisara, Olli Simula y Jari Kangas, eds. *Artificial Neural Networks*. Elsevier Science Publishers, 1991.
- [KNK98] Samuel Kaski, Janne Nikkilä y Teuvo Kohonen. Methods for interpreting a self-organized map in data analysis. En Michel Verleysen, ed., *Proceedings of ESANN'98, 6th European Symposium on Artificial Neural Networks*, págs. 185–190. D-Facto, Belgium, abril 1998. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [KNK00] Samuel Kaski, Janne Nikkilä y Teuvo Kohonen. Methods for exploratory cluster analysis. En *Proceedings of SSRR 2000, International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*. L'Áquila, Scuola Superiore G. Reiss Romoli, agosto–septiembre 2000. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).

- [KO90] Pasi Koikkalainen y Erkki Oja. Self-organizing hierarchical feature maps. En *Proceedings International Joint Conference on Neural Networks IJCNN'90*, tomo II, págs. 279–284. San Diego, 1990.
- [KO95] Rhee M. Kil y Young in Oh. Vector quantization based on genetic algorithm. En *Proc. WCNN'95, World Congress on Neural Networks*, tomo 1, págs. 778–782. Washington D.C., U.S.A., julio 1995.
- [KO98a] Kimmo Kiviluoto y Erkki Oja. Independent component analysis for parallel financial time series. En *Proc. ICONIP'98*, tomo 2, págs. 895–898. Tokyo, Japan, 1998. URL [www.cis.hut.fi/kkluoto/publications/iconip98ica.pdf](http://www.cis.hut.fi/kkluoto/publications/iconip98ica.pdf).
- [KO98b] Kimmo Kiviluoto y Erkki Oja. S-map: A network with a simple self-organization algorithm for generative topographic mappings. En M. I. Jordan, M. J. Kearns y S. A. Solla, eds., *Advances in Neural Information Processing Systems (NIPS'97, Denver, Colorado, USA, Diciembre 2-4, 1997)*, tomo 10, págs. 549–555. MIT Press, Cambridge, Massachusetts, USA, 1998. URL [www.cis.hut.fi/kkluoto/publications/nips97.pdf](http://www.cis.hut.fi/kkluoto/publications/nips97.pdf).
- [Koh72] Teuvo Kohonen. Correlation matrix memories. *IEEE Transactions on Computers*, tomo 21: págs. 353–359, 1972.
- [Koh82] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, tomo 43: págs. 59–69, 1982.
- [Koh86] Teuvo Kohonen. Learning vector quantization for pattern recognition. Informe Técnico TKK-F-A601, Helsinki University of Technology, 1986.
- [Koh88] Teuvo Kohonen. The ‘neural’ phonetic typewriter. *IEEE Computer*, tomo 21(3): págs. 1–22, marzo 1988.
- [Koh90] Teuvo Kohonen. Statistical pattern recognition revisited. *Advanced Neural Computers*, págs. 137–144, 1990.
- [Koh93] Teuvo Kohonen. Things you haven’t heard about the self-organizing map. En *Proc. ICNN'93, Int. Joint Conf. on Neural Networks*, págs. 1147–1156. San Francisco, 1993.
- [Koh98] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, tomo 21: págs. 1–6, 1998.
- [Koh01] Teuvo Kohonen. *Self-Organizing Maps*. Springer, tercera edición, 2001.
- [Koi95] Pasi Koikkalainen. Fast deterministic self-organizing maps. En *Proceedings of International Conference on Artificial Neural Networks ICANN'95*, págs. 63–68. 1995.
- [Kra91] Mark A. Kramer. Nonlinear principal component analysis using autoassociative neural networks. *AIChE Journal*, tomo 37: págs. 223–243, febrero 1991.
- [Kru64] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, tomo 29: págs. 1–27, 1964.

- [Kun99] Sukhamay Kundu. Gravitational clustering: a new approach based on the spatial distribution of the points. *Pattern Recognition*, tomo 32: págs. 1149–1160, 1999. URL [bit.csc.lsu.edu/~kundu/](http://bit.csc.lsu.edu/~kundu/).
- [KVHK97] T. Koskela, M. Varsta, J. Heikkonen y K. Kaski. Time series prediction using RSOM with local linear models. Informe Técnico B15, Helsinki University of Technology, Lab. of Computational Engineering, Espoo, Finland, 1997. URL [citeseer.nj.nec.com/koskela97time.html](http://citeseer.nj.nec.com/koskela97time.html).
- [KVK99] S. Kaski, J. Venna y T. Kohonen. Coloring that reveals high-dimensional structures. En *Data. Proc. of 6th International Conference on Neural Information Processing (ICONIP'99)*, tomo 2, págs. 729–734. IEEE Service Center, Piscataway, NJ, 1999. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [KVK00] S. Kaski, J. Venna y T. Kohonen. Coloring that reveals cluster structures in multivariate data. *Australian Journal of Intelligent Information Processing Systems*, tomo 6: págs. 82–88, 2000. URL [www.cis.hut.fi/~sami/therest.html](http://www.cis.hut.fi/~sami/therest.html).
- [La202] Redes y neuronas: Homenaje a ramón y cajal, 2002. Documental emitido en el programa REDES de La 2 (RTVE), el 3 de noviembre de 2002.
- [LBG80] Yoseph Linde, Andrés Buzo y Robert M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, tomo COM-28(1): págs. 84–95, 1980.
- [Led02] Rubén D. Ledesma. Visualización estadística mediante gráficos dinámicos: Revisión de conceptos, métodos y herramientas informáticas. Informe técnico, Universidad Nacional de Mar del Plata, 2002. URL [www.mdp.edu.ar/psicologia/visual/Defini.htm](http://www.mdp.edu.ar/psicologia/visual/Defini.htm).
- [LF03] Hartmut S. Loos y Bernd Fritzke. Demogng (versión 1.5), 2003. URL [www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html](http://www.neuroinformatik.ruhr-uni-bochum.de/ini/VDM/research/gsn/DemoGNG/GNG.html).
- [Lip87] Richard P. Lippmann. An introduction to computing with neural nets. *IEEE ASSP Magazine*, tomo 4(2): págs. 4–22, abril 1987.
- [LK00] Jouko Lampinen y Timo Kostiaainen. Self-organizing map in data-analysis - notes on overfitting and overinterpretation. En *8th European Symposium On Artificial Neural Networks*. Bruges, Belgium, abril 2000. URL [www.lce.hut.fi/~jlampine/jl\\_bib.html](http://www.lce.hut.fi/~jlampine/jl_bib.html).
- [LSB77] R. C. T. Lee, J. R. Slagle y H. Blum. A triangulation method for the sequential mapping of points from n-space to two-space. *IEEE Transactions on Computers*, tomo 26: págs. 288–292, 1977.
- [Lut90] S. P. Luttrell. Derivation of a class of training algorithms. *IEEE Transactions on Neural Networks*, tomo 1: págs. 229–232, 1990.

- [Ma95] Kristine W. Ma. Applying large vocabulary hybrid HMM-MLP methods to telephone recognition of digits and natural numbers. Informe Técnico TR-95-024, UC Berkeley, International Computer Science Institute, Berkeley, CA, 1995. URL [www.icsi.berkeley.edu/real/rasta.html](http://www.icsi.berkeley.edu/real/rasta.html).
- [Mac67] J. Macqueen. Some methods for classification and analysis of multivariate observations. En L. M. Le Cam y J. Neyman, eds., *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, tomo 1, págs. 281–297. Berkeley: University of California Press, 1967.
- [MADV95] Nasser Mozayyani, Véronique Alanou, J.-F. Dreyfus y Gilles Vaucher. A spatio-temporal data-coding applied to Kohonen maps. En *International Conference on Artificial Neural Networks*, tomo 2, págs. 75–79. EC2 et Compagnie, octubre 1995.
- [Mar93] Thomas M. Martinetz. Competitive hebbian learning rule forms perfectly topology preserving maps. En *ICANN'93: International Conference on Artificial Neural Networks*, págs. 427–434. 1993.
- [MB88] Geoffrey J. McLachlan y Kaye E. Basford. *Mixture Models: Inference and Applications to Clustering*. Marcel Dekker, New York, 1988.
- [MB98] Wolfgang Maas y Christopher M. Bishop, eds. *Pulsed Neural networks*. MIT Press, 1998.
- [MBK98] Ryszard S. Michalski, Ivan Bratko y Miroslav Kubat. *Machine Learning and Data Mining: Methods and Applications*. John Wiley & Sons, Ltd., 1998.
- [MBS93] T. M. Martinetz, S. G. Berkovich y K. J. Schulten. Neural-gas, network for vector quantization and its application to time-series prediction. *IEEE-TNN*, tomo 4(4): págs. 558–569, 1993. URL [www.ks.uiuc.edu/Publications/Papers](http://www.ks.uiuc.edu/Publications/Papers).
- [MC95a] Filip M. Mulier y Vladimir S. Cherkassky. Self-organization as an iterative kernel smoothing process. *Neural Computation*, tomo 7: págs. 1165–1177, 1995.
- [MC95b] Filip M. Mulier y Vladimir S. Cherkassky. Statistical analysis of self-organization. *Neural Networks*, tomo 8: págs. 717–727, 1995.
- [MDB87] B. H. McCornick, T. A. DeFanti y M. D. Brown. Visualization in scientific computing. *ACM Computer Graphics*, tomo 21(6), noviembre 1987. Edición especial.
- [Mic92] Zbigniew Michalewicz. *Genetics Algorithms + Data Structures = Evolution Programs*. Springer Verlag, Varsovia, 1992.
- [Mit95] G. Mitchison. A type of duality between self-organizing maps and minimal wiring. *Neural Computation*, tomo 7(1): págs. 25–35, enero 1995.
- [MJ95] Jianchang Mao y Anil K. Jain. Artificial neural networks for feature extraction and multivariate data projection. *IEEE Trans. Neural Networks*, tomo 6(2): págs. 296–317, marzo 1995.
- [MP43] W. S. McCulloch y W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bull. Mathematical Biophysics*, tomo 5: págs. 115–133, 1943.

- [MP69] M. L. Minsky y S. A. Papert. *Perceptrons: An introduction to computational geometry*. MIT Press, Cambridge, Mass., 1969.
- [MR97] Dieter Merkl y Andreas Rauber. Alternative ways for cluster visualization in self-organizing maps. En *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, págs. 106–111. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, junio 1997. URL [www.ifs.tuwien.ac.at/ifs/research/pub\\_html/mer\\_wsom97/wsom97.html](http://www.ifs.tuwien.ac.at/ifs/research/pub_html/mer_wsom97/wsom97.html).
- [MS91] Thomas Martinetz y Klaus Schulten. A “neural gas” network learns topologies. En Teuvo Kohonen, Kai Mäkisara, Olli Simula y Jari Kangas, eds., *Artificial Neural Networks*, págs. 397–402. Elsevier, Amsterdam, 1991. URL [www.ks.uiuc.edu/Publications/Papers](http://www.ks.uiuc.edu/Publications/Papers).
- [MS94] Thomas Martinetz y Klaus Schulten. Topology representing networks. *Neural Networks*, tomo 7: págs. 507–522, 1994. URL [www.ks.uiuc.edu/Publications/Papers](http://www.ks.uiuc.edu/Publications/Papers).
- [Mur95] F. Murthag. Interpreting the Kohonen self-organizing map using contiguity-constrained clustering. *Pattern Recognition Letters*, tomo 16: págs. 399–408, 1995.
- [Nas92] Guy Philip Nason. *Design and Choice of Projection Indices*. Tesis Doctoral, University of Bath, 1992. URL [www.stats.bris.ac.uk/~guy/Research/PP/PP.html](http://www.stats.bris.ac.uk/~guy/Research/PP/PP.html).
- [NH98] Ari S. Nissinen y Heikki Hyötyniemi. Evolutionary training of behavior-based self-organizing map. En *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence*, págs. 660–665. IEEE, New York, NY, USA, 1998. URL [saato014.hut.fi/Hyotyniemi/publications/98\\_wcci.htm](http://saato014.hut.fi/Hyotyniemi/publications/98_wcci.htm).
- [NTK<sup>+</sup>01] Janne Nikkilä, Petri Törören, Samuel Kaski, Jarkko Venna, Eero Castrén y Garry Wong. Analysis and visualization of gene expression data using self-organizing maps. En *Proceedings of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*. Baltimore, USA, junio 2001. URL [www.cis.hut.fi/sami/mipapers/bioinformatics.shtml](http://www.cis.hut.fi/sami/mipapers/bioinformatics.shtml). Proceedings en CD-ROM.
- [Oja92] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, tomo 5: págs. 927–935, 1992.
- [Par85] David B. Parker. Learning logic. Informe Técnico TR-47, Center for Computational Research in Economics and Management Science, MIT, Cambridge, Mass., 1985.
- [Pat96] Dan W. Patterson. *Artificial Neural Networks: Theory and Applications*. Prentice Hall, 1996.
- [PGJ92] D. T. Pham, P. Garrat y C. Jutten. Separation of a mixture of independent sources through a maximum likelihood approach. En *Proceedings of EUSIPCO, European Signal Processing Conference*, págs. 771–774. 1992.

- [Pol95] Daniel Polani. On the choice of organization measures for self-organizing feature maps. Informe Técnico TR-1/95, Institut für Informatik, Universität Mainz, 1995. URL [www.inb.mu-luebeck.de/publications/publists/Polani-e.html](http://www.inb.mu-luebeck.de/publications/publists/Polani-e.html).
- [Pol97] Daniel Polani. Fitness functions for the optimization of self-organizing maps. En Thomas Bäck, ed., *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*. Morgan Kaufmann, San Francisco, CA, 1997. URL [www.inb.mu-luebeck.de/publications/publists/Polani-e.html](http://www.inb.mu-luebeck.de/publications/publists/Polani-e.html).
- [Pre96] Daryl Pregibon. Data mining. *Statistical Computing and Graphics Newsletter*, tomo 7(3): pág. 8, diciembre 1996.
- [Pri57] R. C. Prim. Shortest connection networks and some generalizations. *Bell Systems Technical Journal*, tomo 36: págs. 1389–1401, 1957.
- [PRM02] Elias Pampalk, Andreas Rauber y Dieter Merkl. Using smoothed data histograms for cluster visualization in self-organizing maps. En *Proceedings of the International Conference on Artificial Neural Networks (ICANN'02)*. Springer, Madrid, Spain, agosto 27-30 2002. URL [www.ai.univie.ac.at/~elias/sdh/](http://www.ai.univie.ac.at/~elias/sdh/).
- [PTVF93] W. H. Press, S. A. Teukolsky, W. T. Vetterling y B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, segunda edición, 1993.
- [PU93] Daniel Polani y Thomas Uthmann. Training Kohonen feature maps in different topologies: an analysis using genetic algorithms. En Stephanie Forrest, ed., *Proceedings of the Fifth International Conference on Genetic Algorithms*, págs. 326–333. San Mateo, CA, julio 1993. URL [www.inb.mu-luebeck.de/publications/publists/Polani-e.html](http://www.inb.mu-luebeck.de/publications/publists/Polani-e.html).
- [Ram77] Santiago Ramón y Cajal. A new concept of the histology of the central nervous system (1892). En D. A. Rottenberg y F. H. Hochberg, eds., *Neurological classics in modern translation*. Hafner Press, New York, 1977.
- [Rau99] Andreas Rauber. LabelSOM: on the labelling of self-organizing maps. En *IJCNN'99. Proceedings of the International Joint Conference on Neural Networks*, tomo 5, págs. 3527–3532. IEEE Service Center, Piscataway, NJ, 1999. URL [www.ifs.tuwien.ac.at/~andi/somlib/labelsom.html](http://www.ifs.tuwien.ac.at/~andi/somlib/labelsom.html).
- [RDG<sup>+</sup>02] Manuel Rubio, Francisco Díaz, Víctor Giménez, Pedro Gómez y Rafael Martínez. Nuevos métodos para análisis exploratorio de datos basado en mapas auto-organizativos de Kohonen. En *1ra. Conferencia Iberoamericana en Sistemas, Cibernética e Informática CИСCI 2002*. Orlando, Florida, julio 2002.
- [RG03] Manuel Rubio y Víctor Giménez. New methods for self-organising map visual analysis. *Neural Computing and Applications*, tomo 12(3-4): págs. 142–152, diciembre 2003. DOI: 10.1007/s00521-003-0387-7.

- [RGD<sup>+</sup>02] Manuel Rubio, Víctor Giménez, Francisco Díaz, Pedro Gómez y Rafael Martínez. Visual analysis of self-organizing maps. En E. Damiani, R. J. Howlett, L. C. Jain y N. Ichalkaranje, eds., *Frontiers in Artificial Intelligence and Applications, KES'2002 Sixth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, tomo 1, págs. 517–521. IOS Press, septiembre 2002.
- [RGG01] Manuel Rubio, Víctor Giménez y Pedro Gómez. Special concept inversion algorithm for advanced cluster analysis in self-organizing maps. En N. Baba, L. C. Jain y R. J. Howlett, eds., *Frontiers in Artificial Intelligence and Applications, KES'2001 Fifth International Conference on Knowledge-Based Intelligent Information Engineering Systems & Allied Technologies*, tomo 2, págs. 1160–1164. IOS Press, septiembre 2001.
- [Rip96a] Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.
- [Rip96b] Brian D. Ripley. Pattern recognition via neural networks. En *A aparecer en un tomo de Oxford Graduate Lectures on Neural Networks, título no decidido*. Oxford University Press, 1996. URL [www.stats.ox.ac.uk/~ripley/papers.html](http://www.stats.ox.ac.uk/~ripley/papers.html).
- [RM99] Andreas Rauber y Dieter Merkl. Automatic labeling of self-organizing maps: Making a treasure-map reveal its secrets. En *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, págs. 228–237. 1999. URL [www.ifs.tuwien.ac.at/~andi/somlib/labelsom.html](http://www.ifs.tuwien.ac.at/~andi/somlib/labelsom.html).
- [RMG86] D. E. Rumelhart, J. L. McClelland y PDP Research Group. *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, tomo 1 y 2. MIT Press, Cambridge Mass., 1986.
- [Ros62] F. Rosenblatt. *Principles of Neurodynamics*. Spartan Books, 1962.
- [RP95] Claudio M. Privitera Réjean Plamondon. A neural model for generating and learning a rapid movement sequence. *Biological Cybernetics*, tomo 74: págs. 117–130, 1995.
- [RS00] Sam Roweis y Lawrence Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, tomo 290(5500): págs. 2323–2326, diciembre 2000. URL [www.cs.toronto.edu/~roweis/lle/publications.html](http://www.cs.toronto.edu/~roweis/lle/publications.html).
- [Rus95] J. D. Russ. *The Image Processing Handbook*. CRC Press, segunda edición, 1995.
- [Sam69] John W. Sammon Jr. A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, tomo c-18(5): págs. 401–409, mayo 1969.
- [Sam70] John W. Sammon Jr. An optimal discriminant plane. *IEEE Transactions on Computers*, tomo c-19: págs. 826–829, septiembre 1970.
- [Sed83] Robert Sedgewick. *Algorithms*. Addison-Wesley, Reading, MA, 1983.

- [SF96] Johan Schalkwyk y Mark Fanty. *The CSLU-C Toolkit for Automatic Speech Recognition*. Oregon Graduate Institute of Science & Technology, Center for Spoken Language Understanding, noviembre 1996. URL [cslu.cse.ogi.edu/toolkit/old/old/version2.0a/documentation/csluc/Cdoc.html](http://cslu.cse.ogi.edu/toolkit/old/old/version2.0a/documentation/csluc/Cdoc.html). Documento Electrónico.
- [She62] R.Ñ. Shepard. The analysis of proximities: multidimensional scaling with an unknown distance function. *Psychometrika*, tomo 27: págs. 125–139 y 219–246, 1962.
- [SM96] Stefan Schünemann y Bernd Michaelis. A self-organizing map for analysis of high-dimensional feature spaces with clusters of highly differing feature density. En *ESANN'96 proceedings - European Symposium on Artificial Neural Networks*, págs. 79–84. D-facto public, Bruges, Belgium, abril 1996.
- [SM00] Stefan Schünemann y Bernd Michaelis. Special algorithms for analysis of clusters with different feature density using self-organizing maps. En *2nd International ICSC Symposium on Neural Computation (NC'2000)*. Mayo 2000.
- [SP98] J. V. Stone y J. Porrill. Independent component analysis and projection pursuit: A tutorial introduction. Informe técnico, Sheffield University, 1998. URL [www.shef.ac.uk/~pc1jvs/papers-published.html](http://www.shef.ac.uk/~pc1jvs/papers-published.html).
- [SSM97] Stefan Schünemann, Udo Seiffert y Bernd Michaelis. Two more modifications of SOMs to handle signals with special properties. En *Proc. of the Workshop on Self-Organizing Maps WSOM'97*, págs. 292–297. Espoo, Finland, junio 1997. URL [iesk.et.uni-magdeburg.de/~seiffert/sffpubl.html](http://iesk.et.uni-magdeburg.de/~seiffert/sffpubl.html).
- [SSS88] W. Siedlecki, K. Siedlecka y J. Sklansky. An overview of mapping techniques for exploratory pattern analysis. *Pattern Recognition*, tomo 21(5): págs. 411–429, 1988.
- [Sta03a] StatSoft, Inc. *Data Mining Techniques*, 1984–2003. URL [www.statsoftinc.com/textbook/stdatmin.html](http://www.statsoftinc.com/textbook/stdatmin.html). Documento Electrónico.
- [Sta03b] StatSoft, Inc. *Neural Networks*, 1984–2003. URL [www.statsoftinc.com/textbook/stneunet.html](http://www.statsoftinc.com/textbook/stneunet.html). Documento Electrónico.
- [Sta03c] StatSoft, Inc. *Selected Topics in Graphical Analytic Techniques*, 1984–2003. URL [www.statsoftinc.com/textbook/stgraph.html](http://www.statsoftinc.com/textbook/stgraph.html). Documento Electrónico.
- [SVSV01] Markus Siponen, Juha Vesanto, Olli Simula y Petri Vasara. An approach to automated interpretation of SOM. En *Workshop on Self-Organizing Maps (WSOM'01), Advances in Self-Organizing Maps*, págs. 89–94. Springer, 2001. URL [lib.hut.fi/Diss/2002/isbn9512258978/](http://lib.hut.fi/Diss/2002/isbn9512258978/).
- [SVVH99] Olli Simula, Juha Vesanto, Petri Vasara y Riina-Riitta Helminen. *Industrial Applications of Neural Networks*, capítulo 4: The Self-Organizing Map in Industry Analysis, págs. 87–112. CRC Press, 1999. URL [www.cis.hut.fi/projects/ide/publications/](http://www.cis.hut.fi/projects/ide/publications/).
- [Teb95] Joe Tebelskis. *Speech Recognition using Neural Networks*. Tesis Doctoral, School of Computer Science, Pittsburgh, 1995. URL [citeseer.nj.nec.com/tebeliskis95speech.html](http://citeseer.nj.nec.com/tebeliskis95speech.html).



- [Tor52] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, tomo 17: págs. 401–419, mayo 1952.
- [TSL00] J. B. Tenenbaum, V. de Silva y J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, tomo 290(5500): págs. 2319–2323, diciembre 2000. URL [www-white.media.mit.edu/~rahimi/markovia/](http://www-white.media.mit.edu/~rahimi/markovia/).
- [Tuk77] John W. Tukey. *Exploratory Data Analysis*. Addison & Wesley, Reading Mass., 1977.
- [Ult92] Alfred Ultsch. Self-organizing neural networks for visualisation and classification. En *Proc. Conf. Soc. for Information and Classification*. Dortmund, abril 1992.
- [Ult93] Alfred Ultsch. Self organized feature maps for monitoring and knowledge acquisition of a chemical process. En Stan Gielen y Bert Kappen, eds., *Proc. ICANN'93, Int. Conf. on Artificial Neural Networks*, págs. 864–867. Springer, London, UK, abril 1993. URL [www.mathematik.uni-marburg.de/~nkisec/pub.html](http://www.mathematik.uni-marburg.de/~nkisec/pub.html).
- [Ult99] Alfred Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. En Erkki Oja y Samuel Kaski, eds., *Kohonen Maps*, págs. 33–46. Elsevier, 1999. URL [www.mathematik.uni-marburg.de/~nkisec/pub.html](http://www.mathematik.uni-marburg.de/~nkisec/pub.html).
- [URLa] URL [www.iro.umontreal.ca/~kegl/research/pcurves/implementations/index.html](http://www.iro.umontreal.ca/~kegl/research/pcurves/implementations/index.html).
- [URLb] URL [lectura.ilce.edu.mx:3000/sites/ciencia/volumen3/ciencia3/130/htm/sec\\_8.htm](http://lectura.ilce.edu.mx:3000/sites/ciencia/volumen3/ciencia3/130/htm/sec_8.htm).
- [URLc] URL [www.kenyon7.freemove.co.uk/membraneexplanation.html](http://www.kenyon7.freemove.co.uk/membraneexplanation.html).
- [URLd] URL [www.uni-kl.de/FB-Biologie/AG-Deitmer/Confocal/gfap3.htm](http://www.uni-kl.de/FB-Biologie/AG-Deitmer/Confocal/gfap3.htm).
- [URLe] URL [www.cimaging.net/Examples/neuronal2.html](http://www.cimaging.net/Examples/neuronal2.html).
- [URLf] URL [www.psych.purdue.edu/Courses/psy320](http://www.psych.purdue.edu/Courses/psy320). Meeting 12.
- [URLg] URL [www.icsi.berkeley.edu/ftp/global/pub/speech/rasta.tar.Z](http://www.icsi.berkeley.edu/ftp/global/pub/speech/rasta.tar.Z). Aplicación Informática.
- [URLh] URL [www.adobe.com/support/techguides/color/colormodels/cielab.html](http://www.adobe.com/support/techguides/color/colormodels/cielab.html).
- [URLi] URL [www.d2.dion.ne.jp/~yoshih/color\\_abc\\_5.html](http://www.d2.dion.ne.jp/~yoshih/color_abc_5.html).
- [URLj] URL [www.cis.hut.fi/sami/ajiips00](http://www.cis.hut.fi/sami/ajiips00).
- [US90] A. Ultsch y H. P. Siemon. Kohonen's self-organizing feature maps for exploratory data analysis. En *Proc. Int. Neural Network Conf. (INNC)*, págs. 305–308. Kluwer, Dordrecht, Netherlands, octubre 1990.
- [UV94] Alfred Ultsch y C. Vetter. Self-organizing-feature-maps versus statistical clustering methods: A benchmark. Informe Técnico Research Report 0994, University of Marburg, FG Neuroinformatik & Kuenstliche Intelligenz, 1994. URL [www.mathematik.uni-marburg.de/~nkisec/pub.html](http://www.mathematik.uni-marburg.de/~nkisec/pub.html).

- [VA95] R. M. Vilar França y B. G. Aguiar Neto. Comparing self-organizing algorithms for vector quantization. En *Proc. International Conference on Engineering Applications of Neural Networks (EANN '95)*. 1995.
- [VA00] J. Vesanto y E. Alhoniemi. Clustering of the self-organizing map. *IEEE-NN*, tomo 11(3): págs. 586–600, mayo 2000. URL [www.cis.hut.fi/~juuso/](http://www.cis.hut.fi/~juuso/).
- [VD01] Thomas Voegtlin y Peter F. Dominey. Recursive self-organizing maps. En N. Allinson, H. Yin, L. Allinson y J. Slack, eds., *Advances in Self-Organizing Maps*, págs. 210–215. Springer, 2001. URL [itb.biologie.hu-berlin.de/~voegtlin/](http://itb.biologie.hu-berlin.de/~voegtlin/).
- [VDG<sup>+</sup>02] Michail Vlachos, Carlotta Domeniconi, Dimitrios Gunopulos, George Kollios y Nick Koudas. Non-linear dimensionality reduction techniques for classification and visualization. En *Proceedings of 8th SIGKDD*, págs. 645–651. 2002. URL [infolab.usc.edu/csci599/Fall2002/paper/DR4\\_kdd02.pdf](http://infolab.usc.edu/csci599/Fall2002/paper/DR4_kdd02.pdf).
- [VDHM97] T. Villmann, E. Der, M. Herrmann y T. Martinetz. Topology preservation in self-organizing feature maps: Exact definition and measurement. *IEEE Transactions on Neural Networks*, tomo 8(2): págs. 256–266, 1997.
- [Ves99] Juha Vesanto. SOM-based data visualization methods. *Intelligent-Data-Analysis*, tomo 3: págs. 111–26, 1999. URL [www.cis.hut.fi/~juuso/](http://www.cis.hut.fi/~juuso/).
- [Ves00] Juha Vesanto. *Using SOM in Data Mining*, abril 2000. URL [www.cis.hut.fi/~juuso/](http://www.cis.hut.fi/~juuso/).
- [Ves02] Juha Vesanto. *Data Exploration Process Based on the Self-Organizing Map*. Tesis Doctoral, Helsinki University of Technology, Espoo, Finland, mayo 2002. URL [www.cis.hut.fi/~juuso/](http://www.cis.hut.fi/~juuso/).
- [VG88] Jacques Vaisey y Allen Gersho. Simulated annealing and codebook design. En *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*, págs. 1176–1179. 1988.
- [VH81] Paul Velleman y David Hoaglin. *The ABC's of EDA: Applications, Basics, and Computing of Exploratory Data Analysis*. Duxbury, 1981.
- [VH02] Juha Vesanto y Jaakko Hollmén. An automated report generation tool for the data understanding phase. En A. Abraham y M. Köppen, eds., *Hybrid Information Systems*, págs. 611–626. Physica-Verlag, Heidelberg, 2002. URL [lib.hut.fi/Diss/2002/isbn9512258978/](http://lib.hut.fi/Diss/2002/isbn9512258978/).
- [VHM97] Markus Varsta, Jukka Heikkonen y José del R. Millan. Context learning with the self organizing map. En *Proceedings of WSOM'97, Workshop on Self-Organizing Maps*, págs. 197–202. Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland, julio 1997. URL [www.cis.hut.fi/wsom97/progababstracts/47.html](http://www.cis.hut.fi/wsom97/progababstracts/47.html).
- [VLM99] A. Vellido, P. J. G Lisboa y K. Meehan. Segmentation of the on-line shopping market using neural networks. *Expert Systems with Applications*, tomo 17: págs. 303–314, 1999.

- [Voe00] Thomas Voegtlin. Context quantization and contextual self-organizing maps. En *Proc. Int. Joint Conf. on Neural Networks*, tomo 5, págs. 20–25. 2000. URL [itb.biologie.hu-berlin.de/~voegtlin/](http://itb.biologie.hu-berlin.de/~voegtlin/).
- [VR95] V. Venkatasubramanian y R. Rengaswamy. *Neural Networks in Chemical Engineering*, capítulo 27: Clustering and Statistical Techniques in Neural Networks, págs. 659–680. Elsevier Science B. V., 1995. (Capítulo de libro invitado).
- [Wer74] Paul Werbos. *Beyond regression: New tools for prediction and analysis in the behavioural sciences*. Tesis Doctoral, Harvard University, Cambridge, Mass., 1974.
- [WF88] J. M. Wolfe y S. L. Franzel. Binocularity and visual search. *Perception & Psychophysics*, tomo 44: págs. 81–93, 1988.
- [WH60] B. Widrow y M. E. Hoff. Adaptive switching circuits. En *IRE Western Electric Show and Convention Record*, tomo 4, págs. 96–104. agosto 1960.
- [Wri77] W. E. Wright. Gravitational clustering. *Pattern Recognition*, tomo 9: págs. 151–166, 1977.
- [Yao99] Xin Yao. Evolving artificial neural networks. *PIEEE: Proceedings of the IEEE*, tomo 87(9): págs. 1423–1447, septiembre 1999. URL [citeseer.nj.nec.com/yao99evolving.html](http://citeseer.nj.nec.com/yao99evolving.html).
- [YB95] Chong-Ho Yu y John T. Behrens. Applications of scientific multivariate visualization to behavioral sciences. *Behavior Research Methods, Instruments, and Computers*, tomo 2: págs. 264–271, 1995. URL [seamoney.ed.asu.edu/~alex/pub/multi-vis/multi-vis.html](http://seamoney.ed.asu.edu/~alex/pub/multi-vis/multi-vis.html).
- [Yu95] Chong-Ho Yu. *The interaction of research goal, data type, and graphical format in multivariate visualization*. Tesis Doctoral, Arizona State University, agosto 1995. URL [seamoney.ed.asu.edu/~alex/education/dissert/dissert.html](http://seamoney.ed.asu.edu/~alex/education/dissert/dissert.html).
- [Yu01] Chong-Ho Yu. *Exploratory data analysis and Data visualization*, 2001. URL [seamoney.ed.asu.edu/~alex/teaching/WBI/EDA.html](http://seamoney.ed.asu.edu/~alex/teaching/WBI/EDA.html). Documento Electrónico.
- [YZ01] Jinsan Yang y B. T. Zhang. Customer data mining and visualization by generative topographic mapping methods. En *Proceedings of the International Workshop on Visual Data Mining*, págs. 55–66. 2001. URL [bi.snu.ac.kr/~scai/Publications/pub\\_dm.html](http://bi.snu.ac.kr/~scai/Publications/pub_dm.html).
- [Zre93] S. Zrehen. Analyzing kohonen maps with geometry. En S. Gielen y B. Kappen, eds., *Proceedings of ICANN'93, International Conference on Artificial Neural Networks*, págs. 609–612. Springer, London, 1993.